In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

3/17/65
b

THE NONSIMULTANEITY CONSTRAINT IN NETWORK-BASED PROJECT MANAGEMENT SYSTEMS

A THESIS

Presented to

The Faculty of the Graduate Division

by

J. Gordon Davis

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Industrial Engineering

Georgia Institute of Technology

May, 1967

THE NONSIMULTANEITY CONSTRAINT IN

NETWORK-BASED PROJECT MANAGEMENT SYSTEMS

Approved:

Chairman

Date approved by Chairman: 5/29/67

ACKNOWLEDGMENTS

To Dr. Adam Abruzzi, the thesis advisor, the author is especially grateful for new perspective in dealing with portions of the research. In addition, Dr. Abruzzi's advice and encouragement were greatly appreciated.

Dr. Robert N. Lehrer, member of the thesis advisory committee, has the author's sincere thanks for his support of the thesis activity and his patient emphasis on progress throughout the research.

Dr. David E. Fyffe, member of the thesis advisory committee, gave valuable assistance in achieving clarity of meaning in the thesis. In addition, Dr. Paul Han, and Dr. Donald O. Covault reviewed the thesis, and for this service the author wishes to express his thanks.

The author's opportunity to participate in a Ph.D. program was made possible by Col. Frank F. Groseclose. His willingness to provide the initial support for the author's studies was greatly appreciated.

The man most responsible for the author's decision to work toward the Ph.D. is Professor Earl P. Martinson, of the University of Florida. There are no words to sufficiently express the author's feeling of gratitude for this individual's influence.

One group of individuals has been especially important to this research. The author's wife, Billie, and sons Michael, Tad, and Griffin, have accepted a situation in which each had to take on responsibilities that normally would have been the author's. For this and for their encouragement they have the author's deepest gratitude.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

Network models of projects may be incorrect because the difference between precedence constraints and nonsimultaneity constraints is either not understood or not appreciated. The nonsimultaneity constraint indicates a relationship among a group of activities such that not all of them can be in progress simultaneously although they are not precedence-related. The incorrectness of a network model can create substantial losses of effectiveness in project planning, scheduling, and control. The purpose of this research is to make explicit the nonsimultaneity constraint and to devise methods by which the activities of a nonsimultaneous set may be optimally sequenced.

The arbitrary or intuitive use of precedence statements to preclude the simultaneous conduct of activities can cause a project schedule to have excessive length. However, the examination of every possible network design which precludes simultaneously becomes computationally prohibitive. The research discussed in this paper provides a method for determining the optimal sequence of the activities in the nonsimultaneous set without the necessity for designing a complete network for every possible sequence.

The essence of the method designed in this research is to remove the nonsimultaneous set from the network and replace it with a set consisting of all nonsimultaneous activities sequenced in all possible ways. Each nonsimultaneous activity is contained in each sequence. The forward

pass procedure is modified and a forward and backward pass made. The maximum criticality in each sequence is calculated and the sequence having the minimum value of maximum criticality is the optimal sequence.

Variations on this basic approach are made to deal with the situation in which there is more than one nonsimultaneous set in a network. The approach involved is a form of dynamic programming. This approach is generalized to deal with situations in which some nonsimultaneous sets will affect the project length and others will not. In this generalized approach, the forward and backward pass allows for the selection of the optimal sequence for one of the initial nonsimultaneous sets. Another forward and backward pass is required for each nonsimultaneous set in the network.

If the nonsimultaneous sets in the multi-set network are independent, a simplified approach allows the optimum sequence for each set to be selected with a single forward and backward pass. The determination of interset dependency is dealt with.

The nonsimultaneity problem also involves sets of activities related such that not all of them may be in progress at one time but more than one may be in progress concurrently. Optimal procedures are given for handling this case when there is only one such nonsimultaneous set in the network or there are multiple independent sets. The dependent multi-set case is discussed but the optimality of the procedure is not proven for this case. Procedures are given for forming the sequences in the case where the simultaneity maximum is greater than one.

Recommendations are made for investigation of other interesting aspects of the nonsimultaneity problem. The question of intersecting nonsimultaneous sets is recommended as suitable for further research. Study is recommended on the evaluation of sequences having unequal direct costs. The design of computer programs to form sequences and evaluate them is recommended. A recommendation is made that further study be undertaken on the situation in which the simultaneity maximum is a function of the activities involved. This would lead to specific resource allocation procedures. Suggestions are made relative to possible sampling procedures which would allow one to use a small fraction of all possible sequences. The result might be a procedure which assigns a priority to the event that the best schedule in the sample is the optimal schedule.

The nonsimultaneity constraint is shown to be of multi-faceted importance. Solutions for the more basic forms of the problem are shown to be much less time consuming with the nonsimultaneity approach than with complete enumeration of all possible network models. Manual approaches are made available for nonsimultaneity sets of small size, say four or less.

GLOSSARY OF TERMS AND ABBREVIATIONS

Basic Network	Network which ignores nonsimultaneity constraint.
Enumerative Network	Network containing alternative sequences of the activities in the nonsimultaneous sets.
Modified Network	Network in which nonsimultaneity is assured by the use of a single sequence for the activities of the nonsimultaneous set.
Precedence-Related	(Activities) connected in a path of "immediately precedes" relationships, thereby precluding the simultaneity of the activities involved.
Forward Pass	Procedure used to calculate ES and EC.
Backward Pass	Procedure used to calculate LS and LC.
ES	Earliest Activity Start Time.
EC	Earliest Activity Complete Time.
LS	Latest Activity Start Time.
LC	Latest Activity Complete Time.
Total Slack	(LC - EC) = (LS - ES); the extent to which activity may be delayed past its ES without causing project completion to exceed its latest allowable time.
Criticality	Negative of Total Slack; (EC - LC) or (ES - LS).
MC	Maximum Criticality over the subscript i.
MMC	$\mathrm{Min}_{j}\ \mathrm{Max}_{i}$ Criticality
X'	The term X calculated on the enumerative network before elimination of any sequences (X' = ES', MC', MMC', etc.).

CHAPTER I

INTRODUCTION

Purpose

The basic purpose of this research is to increase the capabilities of network-based project management systems by improving the efficiency with which the nonsimultaneity constraint can be taken into account when such systems are being used. It is hoped that the explicitness with which this constraint is taken into account in the approaches utilized in this research will lead not only to improved scheduling of projects but also to an improvement in the planning and control phases of project management.

The Nature of Network-Based Project Management

Since 1957, project managers have had access to a new body of philosophy and methodology to aid them in their management function. This new approach involves network representation of projects and has been most closely associated with the names PERT (Program Evaluation and Review Technique) and CPM (Critical Path Method). Many variations of the basic approach exist. It is appropriate at this point to describe only the network technique basic to all these methods.

Definition of a Project

A project may be defined as the activities required to reach a given goal by a given method. It is characterized by the fact that managerial emphasis is focused on the efficiency with which the goal is reached rather than on the efficiency of the individual functions involved in carrying out the project. Activities are subdivisions of the work required to complete the project. Activities are precedence-related if the goal of one activity must be attained before the next one can start. There is no requirement that the activities of a project be precedence-related, but the typical project involves a high degree of precedence relationship among its activities.

Network Representation of a Project

A network is a graphic model of a given project which depicts the activities and their precedence relationships. The network consists of directed lines, called arrows, and nondirected geometrical symbols, called nodes. The model can include other information, such as the duration or resource requirements of the activities. As used herein, the network is assumed not to be time-scaled, i.e., the size and direction of network components have no relation to the time required to carry out any portion of the project represented by the network.

Two different systems of network modeling are in current use. The systems differ in their usage of the symbols composing the network. This difference in usage creates certain significant advantages and disadvantages associated with the use of either system.

Activities-on-Arrows. This system of networking, referred to herein as the A-O-A system, uses arrows to represent activities and nodes to represent events. An event node represents the point in time at which all activities leading into that node are complete. The manner in which the activities are connected through events indicates the precedence relationships among the activities.

Activities-on-Nodes. This system of networking, referred to herein as the A-O-N system, uses nodes to represent activities and arrows to indicate the precedence relationships among the activities. The event concept is not utilized in the A-O-N system. The A-O-N system was selected as being best suited to the conduct and presentation of the research discussed herein.

Mathematical Representation of Network Models

The following system for describing a project and a network model of that project is based in part on work done by Levy, Thompson, and Wiest (12). This system makes possible a concise description of mathematical operations to be performed on the network model.

Let $J = \{J_i\} = A, B, C, \ldots, K$ be the set of activities that must be carried out to complete a project. The symbol $>>$ denotes a relationship such that if $A >> B$, then A is an immediate predecessor of B or, equivalently, B is an immediate successor of A. A project is the set J with all associated $>>$ relationships. The $>>$ relationship requires that A be completed before B can be started.

An activity may have no predecessors, one predecessor, or more than one predecessor. Similarly, an activity may have no successors, one successor, or more than one successor. The set $P_B = \{J_i | J_i >> B\}$, $i = 0, 1, 2, \ldots, k$, where k = the number of immediate predecessors of B, is the immediate predecessor set of activity B. Similarly, $S_B = \{J_i | B >> J_i\}$, $i = 0, 1, 2, \ldots, m$, where m = the number of immediate successors of B, is the immediate successor set of activity B.

An A-O-N network consists of a set of nodes which are in a one-to-one correspondence with the J_i, with an arrow extending from each node representing a member of P_{Ji} to the node representing J_i for all J_i. A convenient convention calls for the addition of two dummy activities, START and FINISH, to the project. START >> J_p for all J_p having no real activity as an immediate predecessor. Thus START becomes the only activity in J having no immediate predecessor. Similarly, FINISH is made the immediate successor of all J_i not having a real activity as an immediate successor. Thus the network starts from a single node and ends at a single node.

A path in the network consists of a subset of J, J_1, J_2, ..., J_n, for which the following holds: $J_1 \gg J_2 \gg \ldots \gg J_n$. The project cannot be completed if there exists any path of the form $J_1 \gg J_2 \gg \ldots \gg J_1$. Such a path is called a loop. A network is acyclic if it contains no loops.

The symbol > retains its traditional meaning when used to relate quantities. When used to relate activity identifiers in an expression such as A > B, it means that there is some path A >> C >> ... >> B. The two-member path A >> B is excluded so that A > B means that A must precede B but must not immediately precede B.

As an example of the use of the system of description given above, consider the project consisting of J = A, B, C, D, E and the relationships A >> C, A >> D, B >> C, B >> D, C >> E, D >> E. START and FINISH are added so that START >> A, START >> B, and E >> FINISH. The A-O-N network is shown in Figure 1 on the following page.

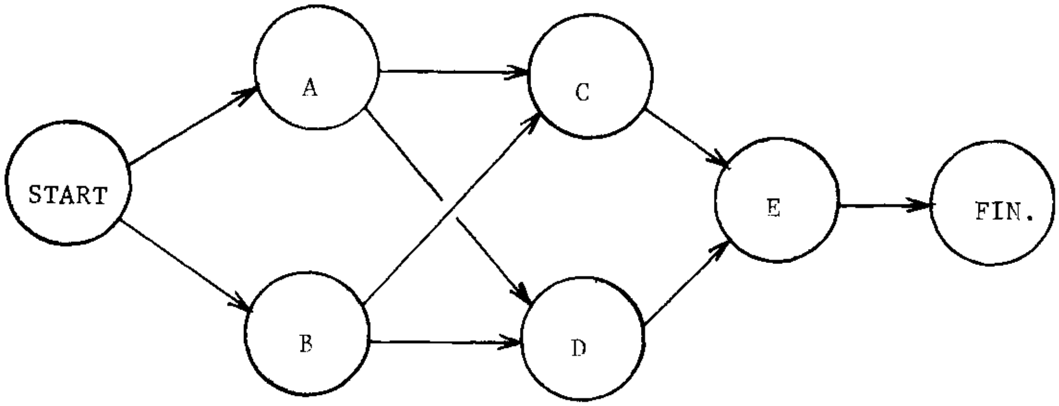


Figure 1. Activity-on-Nodes Network

The immediate predecessor and successor sets are: P_A = {START}, P_B = {START}, P_C = {A, B}, P_D = {A, B}, P_E = {C, D}, P_{FINISH} = {E}; S_{START} = {A, B}, S_A = {C, D}, S_B = {C, D}, S_C = {E}, S_D = {E}, S_E = {FINISH}.

The paths from START to FINISH are: START >> A >> C >> E >> FINISH; START >> A >> D >> E >> FINISH; START >> B >> C >> E >> FINISH; START >> B >> D >> E >> FINISH. There are no loops. The following relationships hold: START > C, START > D, START > E, START > FINISH, A > E, A > FINISH, B > E, B > FINISH.

Network Computations

The time required to complete an activity is known as its duration. The duration of activity A is represented by t_A. The duration of any dummy activity is zero. The basic values sought through network computation are Earliest Start (ES), Earliest Completion (EC), Latest Start (LS), and Latest Completion (LC). The dimension for these values is *working periods from time zero*. Unless otherwise indicated, it is assumed that the ES of START is zero. ES and EC are computed on the basis that each activity J_i is completed in t_{J_i} units of time and is started as soon as all members of P_{J_i} are completed. These computations move from START to FINISH and are referred to as the forward pass.

ES and EC are related such that

$$ES(A) + t_A = EC(A) \tag{1}$$

and

$$ES(A) = \max_{\text{all } J_i \text{ in } P_A} EC(J_i) \; . \tag{2}$$

In making the forward pass, one proceeds along any path from START. Equations (1) and (2) are applied to assign an ES and an EC to each activity. If an activity J_i is encountered such that one or more members of P_{J_i} have not been assigned an ES and an EC, one temporarily defers further consideration of that path. One then finds any other activity whose predecessors have all been assigned an EC but which has not been assigned an ES. One proceeds along any path starting with such an activity until an impasse is reached as cited above or until FINISH is assigned an EC. The above procedure is repeated until it is possible to assign an EC to FINISH. The forward pass is then complete.

LS and LC are related in such a manner that

$$LC(A) - t_A = LS(A) \tag{3}$$

and

$$LC(A) = \min_{\text{all } J_i \text{ in } S_A} LS(J_i) \ . \tag{4}$$

The backward pass moves from FINISH to START, assigning an LC and an LS to each activity, using Equations (3) and (4) and the general approach used on the forward pass.

Total slack, hereinafter referred to simply as slack, is defined as the number of time units by which an activity may have its completion delayed past its EC without affecting the LC of FINISH. The following equation is used in computing the slack associated with activity A.

$$S(A) = \text{Slack}(A) = LC(A) - EC(A) = LS(A) - ES(A) \tag{5}$$

Free slack is defined as the number of time units by which the completion of an activity can be delayed past its EC without affecting the ES of any of its immediate successors. The equation for the calculation of free slack associated with activity A is

$$\text{Free Slack}(A) = \min_{\text{all } J_i \text{ in } S_A} ES(J_i) - EC(A) \tag{6}$$

The criticality of activity A, C(A), is defined as the negative of the slack of A.

$$C(A) = -S(A) \tag{7}$$

All activities J_i for which $C(J_i) = C(\text{FINISH})$ constitute a subset of J from which the critical path or paths are formed. C(FINISH) is the criticality of the project. LC(FINISH) may be based on constraints arbitrarily imposed on the project and may thus be equal to, greater than, or less than EC(FINISH). Thus C(FINISH) may be zero, negative, or positive. Unless otherwise stipulated, C(FINISH) is taken to be zero. When C(FINISH) is zero, all activities having zero criticality are on the critical path.

The length of path T_i is represented by $L(T_i)$ and is defined as the sum of the durations of the activities composing the path. Thus if $J_1 >> J_2 >> \ldots >> J_n$ = path T_1, $L(T_1) = \sum_{i=1}^{n} t_{Ji}$. If $L(T_1)$ is equal to or greater than all other $L(T_i)$ for the network, then T_1 is the critical path or one of the critical paths. If the $\max_i L(T_i)$ is common to two or more paths, it is possible that some activities may be common to more than one critical path.

<u>The Phases of Network-Based Project Management</u>

A distinct separation exists between the planning phase, the scheduling phase, and the control phase of project management when a network-based approach is used. The separation of the planning phase and the scheduling phase constitutes one of the major advantages of network analysis over bar-chart techniques. The emphasis on the separation of the control phase from the other two phases is primarily to focus attention on what tends to be a neglected aspect of project management.

<u>The Planning Phase</u>. In this phase of project management, specifications for the project goal are designed and all parties to the

project verify that no ambiguities or omissions exist in these specifications. Project constraints are determined. These constraints include such factors as imposed earliest or latest starts or completions on any activities of the project, particularly START and FINISH; maximum availability of various types of resources required by the activities; periods of time during which certain types of activities cannot be in progress; indirect costs, including bonuses or penalties associated with the length of the project; and all criteria which will be used in evaluating alternative schedules.

The activities and precedence relationships are determined in the planning phase and the network representing them is constructed. As there may be several different ways of achieving the project goal, this phase may involve the design of several networks, the best of which can be determined only after the scheduling phase has been completed for each. Each network must represent a specific method of carrying out the project. The estimated duration of each activity is determined at this time. These duration estimates should be accompanied by the resource types and levels which the estimator has in mind when making each estimate. As activities generally may have their rate of progress varied by variations in the levels of resources applied, it is often desirable to obtain several duration-resource level estimates for each activity. The estimate to be used in the initial schedule calculations would be the one associated with the resource level which is most efficient for carrying out the activity when that activity is considered independently of any others that might be in progress at the same time. In some instances, it is desirable to determine the direct costs of an activity instead of, or as a supplement to, specific resource types and levels.

The Scheduling Phase. The scheduling of activities involves the decision as to when an activity should be started and when it should be completed. Some of the questions to be asked in judging the desirability of a given schedule are the following:

(1) Does the schedule minimize the sum of the direct and indirect costs?

(2) How "smooth" is the resource requirement schedule?

(3) How closely does the resource requirement schedule adhere to stated maximum resource availabilities?

(4) Does the schedule minimize the project length?

(5) Is the project length short enough to meet the deadline?

A forward and a backward pass are usually the first steps in scheduling. Many times a schedule with all activities at their ES or LS will meet all criteria established for scheduling. Oftentimes, however, it will be necessary to shift activities, change the resource levels applied to activities, or interrupt activities in order to meet the established criteria.

In scheduling a project, it may be necessary to lengthen the project to a duration greater than the length of the critical path. In such cases one may find that the project length depends not on the critical path, but rather upon the availability of certain types of resources. Unfortunately, much of the basic literature on network-based techniques discusses project length without giving consideration to resource availability.

The Control Phase. After an acceptable schedule has been designed, it is necessary to have a control system which will provide the basis for adjusting the schedule throughout the life of the project. Such adjustment is usually necessary in order that the schedule may continue to meet the criteria established for it. If no adjustment is necessary, it is still necessary to have a control system to verify that such is actually the case.

The control system should provide first of all for the flow of progress information from the project to the project manager. Another flow of information which also must be provided for is that concerned with changes in the project plan. The project manager also may change activity duration estimates and resource requirements, based on his comparisons of actual quantities with estimated quantities in these two categories. The control system should provide for replanning and rescheduling based on the new information available. Lastly, the control system should provide for the updating of all project documents so that all individuals concerned are using the latest schedule.

The Nonsimultaneity Constraint

The precedence relationships which the network can show as existing between activities A and B are $A \gg B$, $A > B$, $B \gg A$, or $B > A$. If none of these relationships holds, the network implies that A and B could be in progress at the same time, although the network never indicates that A and B *should* be in progress at the same time. However, even when A and B are not precedence-related, there may be some reason why they should not be in progress simultaneously. This nonsimultaneity constraint leads to a schedule in which either A is completed before B starts or vice versa.

There are many possible reasons for the existence of the nonsimultaneity constraint. The most common reason is the restricted availability of resources mentioned earlier. Safety factors can constitute another reason for precluding the simultaneous conduct of activities. A third reason for this constraint may be that the environmental conditions associated with one activity make the simultaneous satisfactory conduct of some other activity difficult or impossible. Sandblasting or spray painting are activities which might fall in this category. Limited space in which to work can lead to a nonsimultaneity restriction. The limited availability of supervision may be the reason for the existence of the constraint. All of these reasons can be broadly conceptualized as limited resource problems, but fitting some of them into a specific limited resource problem may be quite difficult. It is desired herein not to restrict the consideration of this research to the problem arising from any specific reason for the existence of the nonsimultaneity constraint, but to deal only with the fact that there does exist a stated degree of nonsimultaneity.

The Importance of the Nonsimultaneity Problem

The Measure of Effectiveness of Nonsimultaneity Resolution

Regardless of the means by which one deals with the nonsimultaneity constraint, the result will be a sequencing of the activities involved as if there were additional precedence relationships added to the network. The sequence selected for the nonsimultaneous activities will affect the ES of at least a portion of the succeeding activities. It is possible that the ES of FINISH will be affected by the sequence selected. Minimization of project length is used as a primary criterion by which alternative sequences of nonsimultaneous activities are evaluated. A constant, LC(FINISH), is used in evaluating alternative sequences; consequently, the criterion of minimization of project length is equivalent to minimizing the criticality of FINISH.

In some networks, it is impossible to affect the project length by the sequence of nonsimultaneous activities chosen. This does not mean that the choice of sequence is unimportant, however. In general, some sequences will create higher criticality than others, even though this criticality is a negative figure. A small negative criticality value indicates an activity which could experience little delay in starting without becoming a member of the critical path and thus affecting project length. For this reason, the measure of effectiveness used in evaluating alternative sequences of nonsimultaneous activities is the maximum criticality, MC, among the activities affected by the sequencing. The basic criterion used is the minimization of MC among the activities affected. This criterion is dealt with further in Chapter III.

The Computational Magnitude of the Problem

Assuming that the selection of a sequence for evaluation is the equivalent of adding the precedence requirements between the members of the nonsimultaneous set in such a manner that no other sequence is possible, the optimal sequence can be selected by evaluating all of the activities in each of the networks resulting from the various ways in which the nonsimultaneous activities can be sequenced.

Let a basic network be defined as a network in which no features have been added to resolve the nonsimultaneity conflicts. If there is only one nonsimultaneous set in the basic network and this set contains only two activities, there are only $2! = 2$ networks to examine. If, however, there are two nonsimultaneous sets in the network, each set containing six members and being so constrained that no two members of a set can be in progress at one time, there are $(6!)^2 = 518{,}400$ networks to examine.

The magnitude of the computational problem is a function of both the extent of the nonsimultaneity constraint and the total number of activities in the basic network. Complete enumeration of all possible networks in which nonsimultaneity is assured through the addition of the minimum number of precedence relationships is not generally feasible. The computational magnitude of the problem has tended to create acceptance of intuitive solutions rather than solutions of proven optimality or near-optimality.

The Importance of the Optimal Solution

Giffler, Thompson, and Van Ness (20) discuss a Monte Carlo approach to the selection of a production schedule where n commodities are to be processed over one or more of n facilities. The basic value of their work lies in the concept of generating a sufficient number of feasible schedules to make the probability that the best of these is an optimal schedule sufficiently high. The binomial approach is used, where p is the probability that a schedule generated by a given set of facility loading rules will be an optimal schedule. Then the probability of getting an optimal schedule in n trials is $1 - (1 - p)^n$.

If this concept is applied to the random resolution of nonsimultaneity problems, we find that we can calculate the probability that the best of a group of schedules is an optimal schedule. For example, consider a schedule that has no simultaneity conflicts because the nonsimultaneous activities were put in sequence in a random fashion. If we know that the probability that any such schedule is optimal is 0.01 and we further determine that this schedule is the best of 200 similarly generated schedules, then the probability that this schedule is optimal is 0.866. However, if the probability that any such schedule is optimal is 0.002, the best of 200 such schedules has a probability of only 0.330 of being the optimal schedule.

It is obvious that the validity of the above approach is directly related to the accuracy and precision of the estimate of the probability that a schedule selected at random from a group of randomly generated schedules is an optimal schedule. For this approach to have practical value, it is essential that means be available to base such estimates on readily observable characteristics of the project, such as number of activities in the network, number of nonsimultaneous sets in the network, complexity of the nonsimultaneity constraint within sets, variation in activity durations within nonsimultaneous sets, and such other factors as are found to significantly influence the probability being estimated. Assuming that analytical approaches to calculating this probability from data on the factors mentioned above are not available, some means of generating an optimal schedule for a given set of factor levels is required. Then the ratio of optimal schedules to total schedules in a sample of randomly generated schedules can be used as an estimate of the probability in question for the set of factor levels in effect for the sampling. The extent to which the effects of varying factor levels can be experimentally examined is in part dependent upon the ease with which an optimal schedule can be generated. An approach to generating an optimal schedule might be used effectively in such experimental work even though the approach is too involved for most practical applications.

In the scheduling of actual projects, one finds that some projects require only a feasible schedule which does not cause a target date to be overrun. On other projects, however, one finds situations in which a schedule differing only slightly from the optimal schedule might be much more costly than the optimal. In such cases, the availability of a technique for designing an optimal schedule with much less effort than is required for complete enumeration of all schedules would be of interest.

The Effect of the Nonsimultaneity Problem on Project Planning

When the network analyst obtains precedence information from the project manager, one of the most difficult tasks the network analyst has is ascertaining that the precedences being stated are not attempts to resolve nonsimultaneity problems. Maximum flexibility in scheduling requires that precedence requirements indicated in the basic network be technological in nature. This means that the goal of a technological predecessor must be reached before the work of a successor activity can be started. If the project manager recognizes that two activities cannot be in progress simultaneously, he may attempt to solve the problem by imposing a precedence requirement between the activities although the activities are not technologically precedence-related.

Such decisions about sequence are in reality attempts to carry out the scheduling function before sufficient information relative to the consequences of these decisions is available. These intuitive decisions may preclude consideration of the best solutions to the problem; for example, the optimal sequence for a nonsimultaneous set may be the one which the project manager would intuitively classify as the least desirable. However, if an explicit method of dealing with nonsimultaneities is available, the project manager will have less tendency to attempt to resolve such conflicts by precedence statements.

Explicit differentiation between precedence-related activities and nonsimultaneous activities may cause the project manager to find that what he had previously regarded as precedence requirements are in fact traditional means of precluding simultaneity. The network analyst generally will not be able to detect that such precedence requirements are improper. Thus, acquainting the project manager with the nature of the non-simultaneity constraint and assuring him that the procedures being employed will take them into account offer more promise that the basic network will indicate only technological precedence than if technological precedence is explained but nonsimultaneity is not.

The Effect of the Nonsimultaneity Problem on Project Control

In the control phase of project management, revisions in the schedule for the remaining activities are likely to be made at each updating. Even though the original schedule contained optimal sequences of nonsimultaneous activities, there is no guarantee that these sequences are optimal at the time of updating. If these sequences are not indicated as arising only from nonsimultaneity constraints, they are likely to be considered as true precedence requirements after the initial scheduling is done. It is necessary that alternative sequences be available for consideration at every rescheduling if an optimal schedule is to result. Therefore, procedures tending to keep the nonsimultaneity problem explicit throughout the life of the project tend to lead to better overall conduct of the project than more traditional means of dealing with nonsimultaneity.

Scope and Limitations of the Research

The research described herein deals with the nonsimultaneity problem in project scheduling without regard to the reasons for the nonsimultaneity constraint. Recognizing that network-based techniques are widely accepted in the project management field and that they represent the best currently available approaches to project management, the author restricted this research to procedures which can be implemented by using existing networking technology. Production scheduling problems are not considered, though it is likely that some of the results obtained herein can be incorporated into production scheduling techniques.

The projects considered in this research consist of activities having fixed durations for a given scheduling attempt. An activity, once started, must continue to completion without interruption. All resource requirements and limitations are taken into account in the stated nonsimultaneity requirements. The only scheduling criterion is minimization of criticality among the activities affected by the sequencing of nonsimultaneous activities.

Multiple as well as single sets of nonsimultaneous activities within a given network are treated. Varying degrees of simultaneity within a set are examined. These degrees range from the situation in which only one member of a set may be in progress at any one time to the situation in which all but one member can be in progress at any one time. Also treated is the case in which the number of activities from a given set which can be in progress simultaneously varies according to the identity of the activities. Approaches to the problem of intersecting nonsimultaneous sets are given.

Comparisons of magnitude of calculations are made between the approaches designed in this research and complete enumeration and between the approaches described herein and existing approaches to incorporation of the nonsimultaneity constraint into the scheduling procedure. Various possibilities for compromising between existing approaches and the major approach outlined in this research are investigated.

Objectives

The prime objective of this research is the development of a methodology for optimally resolving the nonsimultaneity problem without examining every possible schedule which achieves this resolution. Secondary objectives are the improvement of nonoptimal techniques for dealing with simultaneity constraints and the improvement of techniques for analyzing project networks.

CHAPTER II

LITERATURE SURVEY

Introduction

The literature cited herein deals primarily with network-based approaches to the solution of the resource allocation form of the nonsimultaneity problem. In addition, representative literature is cited dealing with the general network-based approach to project management. Some early developmental work in project management leading up to the advent of network-based techniques also is discussed. A few additional items are included because of their possible adaptation to the problem area under consideration.

Early Work in Managing Complex Work Efforts

Among the first to formally treat the problem of constraints in activity scheduling was Taylor (23), who advocated a planning department for the production function. This department was to sequence and schedule activities in an essentially job-shop environment. Taylor published no specific procedures for carrying out this management function, but he did take note of the limited resource constraint when he indicated that a running balance of the future work planned for each class of men and machine should be maintained.

Gantt (4) apparently was the first to introduce a specific technique for the planning and scheduling of work. His technique involved a graphic representation of scheduled activities, each activity being represented by a line or bar whose length was proportional to the amount of time allocated for that activity. The activity symbols were placed on a time-scaled chart so that the scheduled start and completion of each activity could be noted graphically. An accompanying chart was used to show the extent to which production facilities were loaded by future work. The bar chart approach, widely referred to as Gantt charting, still is a popular technique in both production and project management.

Knoeppel (10) introduced a graphic technique which has the effect of making explicit the precedence relationships which exist among activities but are only implicit on the Gantt chart. While Knoeppel apparently restricted his use of this approach to time-scaled representations of the filling of production orders, with all activities shown at their latest times consistent with the promised delivery date, he actually was the originator of the network representation of a set of interrelated activities. Knoeppel's approach failed to gain appreciable recognition, while the Gantt chart was widely used, even though the Knoeppel chart was capable of showing everything shown on the Gantt chart plus the restricting precedence relationships.

The work of Taylor, Gantt, and Knoeppel was done in the period around the turn of the century. Their ideas appear to be the last new ideas on planning and scheduling for nearly 50 years. Koepke (11) presented planning and scheduling techniques in 1949 which represented no improvement over those of Gantt and Knoeppel. At about this time, the

Line of Balance technique (13) was introduced. It is basically a Knoeppel chart, with an added control feature which shows actual activity progress in comparison with planned progress, both in absolute units and in percentage of total activity requirements. Designed for management of large production orders, it is heavily oriented toward accounting for the number of individual units which have been processed through a certain activity.

The Emergence of the Project Management Function

Baumgartner (1) describes a trend in the late 1950's toward project orientation and the growth of project organizations within functional organizations. He cites as factors responsible for this trend such things as rapid technological advance and the change in theories and philosophies of national defense and prestige. The increased complexity of projects in the nuclear and space age and the increased emphasis on speed in achieving goals have put heavy demands on the project management field to provide the techniques and philosophies necessary to cope with an increasingly difficult task. Muth and Thompson (20) describe another aspect of technological advance which has had a profound influence upon the project management field:

> In recent years there has been a great increase in interest in industrial scheduling problems from the points of view of both practice and research. Of course, such problems have existed since the creation of the first factory consisting of more than one worker or machine. But the solution of these problems by hand is impossible. Hence the recent interest in their solution has centered on attempts to make use of electronic computers for solving them.

Network-Based Project Management Systems

Documentation of the first network-based project management systems appeared in 1958. These two systems, PERT (15) and CPM (25), were developed independently and, to a large extent, concurrently. The terms PERT and CPM have since lost their specific identities and now are used interchangeably to refer to a network-based project management system. Later versions of the original systems include PERT/COST, PERT/RELIABILITY, a large number of resource allocation schemes, and a few specialized applications of the networking concept.

PERT (Program Evaluation and Review Technique)

PERT originally was designed to deal with projects having activities whose durations are subject to random variation. The basic PERT model uses knowledge of this variability to make statements regarding the probability of the project's reaching a given stage of completion by a given time. An integral part of this probability approach is a system of estimating activity durations. This system, called the three-time estimate system, requires an estimate of *optimistic, pessimistic,* and *most likely* time for each activity. Though he may not wish to make probability statements, the three-time estimate system can be used to allow the conscientious estimator to express his uncertainty about his estimates of activity durations. Moder and Phillips (18) revised the definitions of optimistic and pessimistic to forms which are statistically and practically more desirable than the original definitions.

CPM (Critical Path Method)

CPM was designed originally for projects where activity times are essentially deterministic for a given level of expenditure of resources, but this level can be varied by varying the amount of money spent for direct cost factors. A given change in resource level, and hence direct cost, results in a given change in activity duration. The limits for time and cost are called *crash* and *normal*. Crash time is the shortest duration possible for the activity in question and crash cost is the minimum direct cost at which crash time can be achieved. Normal cost is the minimum direct cost at which the activity can be carried out. Normal time is the minimum duration possible without exceeding normal cost. If all activities are at normal, the project length can be reduced optimally by finding the critical path activities which can be shortened in duration at the least increase in direct costs. In this manner, a relationship can be developed between project length and total project direct costs. Combining this information with knowledge of the relationship between indirect project costs and project length allows the project manager to determine the optimal project length and the optimal schedule to achieve that length.

Kelley (9) describes the mathematical basis for finding the optimal schedule. This basis involves the parametric linear programming model, its relationship to the primal-dual algorithm, and a network flow algorithm for solving the network flow problem fundamental to the primal-dual algorithm.

Clark (3) deals with the CPM problem by means of a system which leads to optimum small augmentations of resources to reduce the project length. The combined augmentation is not necessarily optimum.

PERT/COST

The PERT/COST approach (5) does not provide probability information relative to costs, as one might suspect from the title, but is primarily designed to aid in project cost control. The project orientation trend mentioned earlier has forced the accounting function to look at the problem of accumulation of cost information across functional areas so that project cost control can be carried out. The PERT/COST approach provides a means of using the basic PERT information system to combine time and cost data for more meaningful project control.

PERT/RELIABILITY

PRISM (Program Reliability Information System for Management) and the two approaches followed under this system are discussed by Frambes (7) and Malcolm (14). One approach, RPM (Reliability Performance Measure), is concerned with providing a prediction, in the form of a probability statement of fraction successful, of the eventual operational reliability of the end item and its subsystem components to be made at each stage of the development cycle. The other approach, (RMI (Reliability Maturity Index), provides a running measure of the compliance with planned reliability activities by collecting, analyzing, and displaying information on the progress of the reliability documentation program and the quality and significance of the documents produced.

Resource Allocation Methods

None of the project management systems mentioned above explicitly take into account the problem of availability of resources required to comply with a given schedule. If resources are not unlimited, however, the major problem may be the allocation of resources to activities in a manner such that scheduling criteria may be met.

Kelley (8) gives an analysis of the resource allocation problem and some feasible approaches to its solution. He states that the size of the combinatorial problem involved in formulating and solving the resource loading problem appears to preclude any direct approach. Basically, according to Kelley, the problem involves scheduling the activities of a project so that (1) the work is performed according to the method and in the sequence planned, (2) the resources required do not exceed assumed availabilities, and (3) the duration of the project is minimized. He states that it is neither easy nor necessarily practical to satisfy condition (3) exactly and that one may, for most practical purposes, substitute the less exacting condition that the duration of the project be reasonably near the minimum.

Kelley divides resource allocation techniques into serial methods and parallel methods. The input to the serial method is a list of all the activities in a project ordered in such a manner that no activity appears before all its predecessors have appeared. However, many different orderings may meet this criterion, so other, lower-level, criteria may be employed to order activities which are not ordered by the predecessor criterion. These secondary criteria may include criticality, duration, dollar value, quantity of resources required, or some subjective measure of importance.

The serial method schedules one activity at a time, starting with the first on the list. The scheduled start of the activity is the first time at which all its predecessors are complete and sufficient resources are available. No activity is considered until the activity immediately before it in the list has been scheduled. Kelley indicates that the possibility of splitting activities (stopping work on an activity before its completion and resuming work at a later time) should be considered a part of the basic form of the serial methods. He also discusses the problem of starting an activity with less than the stated resource requirements.

Parallel methods examine groups of activities and lead to decisions as to which of the activities in a group should be started at the point in time, t, for which the scheduling decision is being made. Any activity in the group which cannot be started at time t becomes part of another group which will be considered at some later point in time. The members of the group at time t include all activities which have their predecessors completed. The activities selected from that group for starting at time t are chosen by the application of an arbitrary decision rule. Such a rule might be the following: Order the activities in the group by decreasing criticality. Start scheduling the activities to start at the time in question, beginning with the most critical activity and proceeding through the criticality ordering. If insufficient resources are available at time t to start an activity, it is not scheduled with this group. The scheduling at time t is complete when every member of the group has been considered. The time then advances to the next time at which an activity is completed. The scheduling process is then repeated at this time. The process continues until all activities are completed.

Kelley feels that serial methods are more practical than parallel methods. He says that although it can be argued that parallel methods take into account more information at each step and therefore ought to give better results, the hope is illusory.

Mauchly (17) points out that schedules generated by the basic CPM approach may prove to be quite unworkable unless any existing restrictions on resources are taken into account. He advocates an approach which initially schedules each activity at its earliest start, ignoring resource restrictions. The resulting resource requirement schedules are examined for excessive quantities. Intuitive shifts of activities, utilizing existing slack first, then considering increasing the length of the project, are made until no excessive resource requirements exist.

> Thus, by a combination of intuition and a series of successive approximations, the analyst is able to come up with a solution to a combinatorial problem that cannot be solved mathematically in a reasonable length of time. Although the solution is not mathematically derived, it is accurate enough for all practical purposes, and far more useful than any at which he could arrive by relying on intuition alone.

Martino (16) proposes a procedure called MAP (Multiple-Resource Allocation Procedure) which utilizes what he terms a dynamic serial-parallel precedence procedure in assigning priorities to activities competing for limited resources. This procedure gives priority at any assigned time to the activity with the least slack. The scheduling procedure starts at time zero and proceeds as follows:

Step One. All activities which have their predecessors completed are considered for starting at this time. The eligible activities are considered in descending order of criticality. Each activity in its turn is scheduled to start at this time if sufficient resources exist. After all eligible activities are considered for possible start at this time, the scheduling procedure advances to the next point in time at which either additional activities become eligible for scheduling consideration or additional resources become available or both conditions occur.

Step Two. The criticality of the remaining activities may be changed by the scheduling of Step One; therefore, the criticality of activities which are eligible for scheduling at this new point in time is calculated and Step One repeated.

Step Three. Steps One and Two are repeated until all activities are scheduled.

In assigning scheduling priorities, Martino uses criticality as the prime criterion but, in case of a tie, resorts to the following priority ordering rules in the order listed until the tie is broken:

(1) Descending order of need of overall resources (number of resource-time units)

(2) Descending order of crew size (number of resource units)

(3) Increasing order of sequence code (successor event number)

He mentions that "latest start" can be used as the primary criterion and that so doing has the apparent advantage of not requiring a recalculation of priority at each point in time at which scheduling decisions are made. He then dismisses this idea by saying that this apparent advantage is of no value when negative slack is encountered. However, Moder and Phillips (18) use latest start as the criterion for assigning priority in the limited resource problem and achieve the same results as MAP gives with far less computation and a much more easily applied technique. Martino's reason for not using latest start as the criterion is not valid, for decreasing order of current criticality of activities whose predecessors are complete gives exactly the same arrangement of those activities as does increasing order of their latest start values. Criticality must be recalculated after each movement to a new scheduling time, but latest start values from the original backward pass can be used for the entire scheduling process. The relative values of latest start of the unscheduled activities remain unchanged when the latest completion of the project is changed through scheduling of the activities competing for limited quantities of resources.

Verhines (24) refutes by example the claim that criticality is the optimal basis for establishing priorities for the assignment of scarce resources. Figure 2 is the network of a project which must be scheduled with only one repairman.

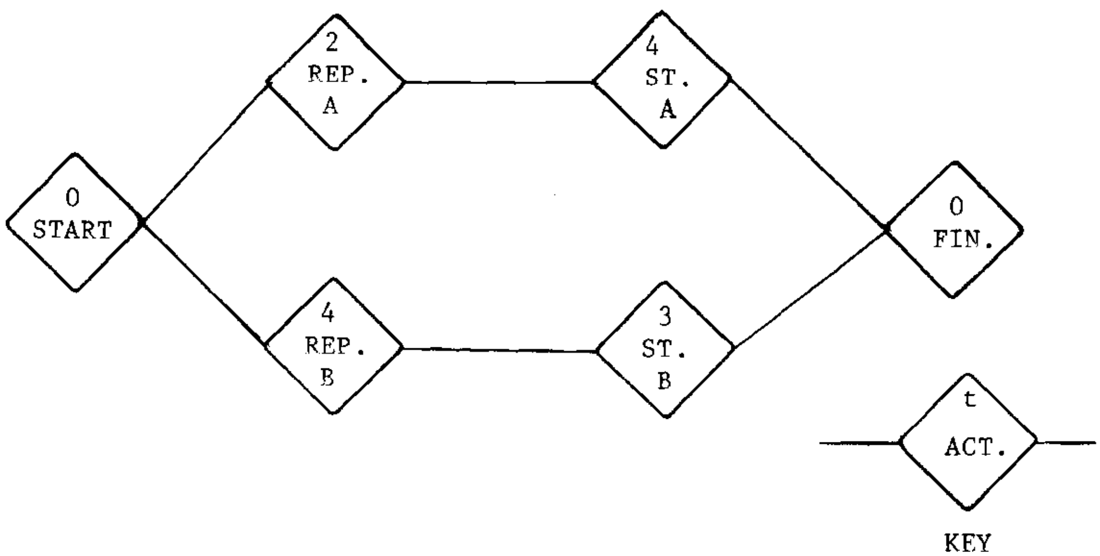


Figure 2. Verhines' Example Project

This one man must repair both A and B. The B path is more critical, but if B is repaired first, the project will take ten time periods. If the noncritical activity, "Repair A," is scheduled to be done first, however, the project can be completed in nine time periods.

Verhines here is dealing with the situation in which a group of activities having all their predecessors complete are competing for a supply of resources insufficient to allow all of them to be started at the same time. He states that assigning priority on the basis of longest remaining series of activities (which is the equivalent of using LC) results in the shortest overall project completion time. He further suggests that activities having equal priority under this rule should be started in order of increasing activity duration.

The approaches of previously cited authors all ignore the fact that it may be desirable not to schedule an activity to start at a given point in the scheduling procedure although all its predecessors are complete, it has scheduling priority, and the necessary resources are available. The network in Figure 3 on the following page depicts a project in which such a situation arises. Activities C_1 and C_2 each requires one unit of resource type X and only one unit of resource Type X is available. Most approaches to allocation of the one unit of type x resource would stop at time three, examine the activities to see which had all their predecessors complete, see that C_2 was the only candidate for starting at that time, see that sufficient resources were available, and therefore schedule C_2 to start at time three. If the activities cannot be split, starting C_2 at time three means that C_1 cannot start until time eight although its predecessor is completed at time five. The resulting project length is 20 time units. If, however, C_2 is not started until C_1 is completed, a project length of 17 time units is realized.

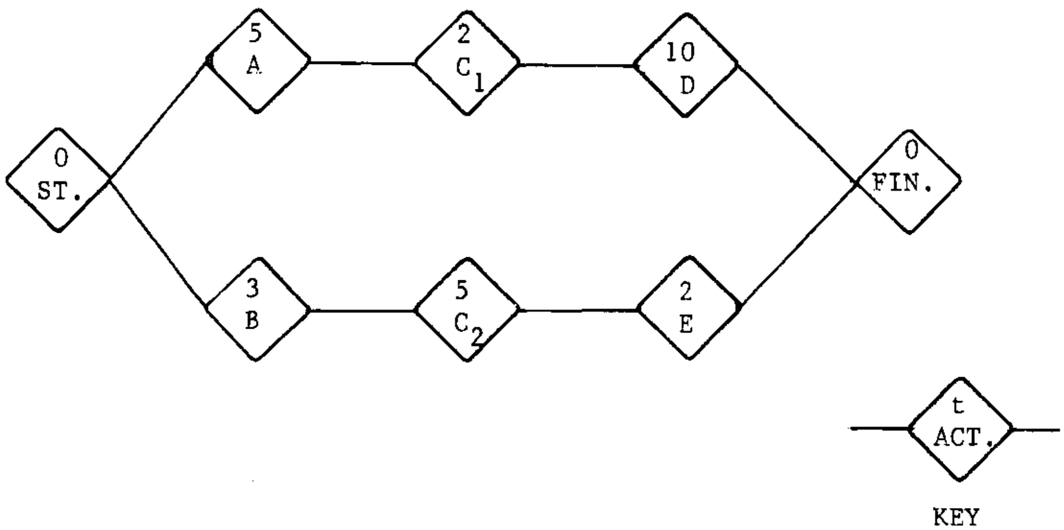


Figure 3. Project Demonstrating Situation Not Discussed by Previously Cited Writers

To a limited extent, Shaffer, Ritter, and Meyer (22) take into account the problem cited above. They approach the problem of resource allocation by starting with a schedule which has been generated without violating the precedence restrictions but results in required resource levels higher than those acceptable to the project manager. Their approach, RSM (Resource Scheduling Method), is a process which makes adjustments to such an unacceptable schedule so as to reduce the maximum resource requirements to levels acceptable to the project manager while increasing project length by a minimal amount.

RSM utilizes the addition of pseudo-precedence arrows to the network to preclude the simultaneous conduct of activities where such simultaneity would create unacceptably high levels of resource requirement. The procedure calls for the calculation of EC(I) - LS(J) for all pairs of activities I and J which, if put in sequence by adding I >> J to the precedence requirements, would reduce the resource requirements to an acceptable level. The I,J pair for which EC(I) - LS(J) is a minimum is selected as the pair which will give a minimum increase in project length.

The designers of RSM fail to indicate how one would deal with the situation in which it would be necessary to put more than two activities in sequence in order to satisfactorily reduce the resource requirements. It also ignores the effect of the sequence selected at one stage of the scheduling process upon the sequence to be selected at another stage. Figure 4 on the following page depicts a project for which RSM would give the optimal answer at each scheduling stage, but in so doing would create a suboptimal overall schedule. Under this approach, $EC(A_1) - LS(A_2) = 2$ and $EC(A_2) - LS(A_1) = 0$, so A_2 is made to precede A_1. Using the new $EC(A_1)$, a new ES and EC are calculated for B_1. With these new values, $EC(B_1) - LS(B_2) = 6$ and $EC(B_2) - LS(B_1) = 4$. This indicates that B_2 should precede B_1. The resulting overall schedule results in a project length of 15 time units. If, however, the added precedence requirements had been $A_1 >> A_2$ and $B_1 >> B_2$, the project length would be 13 time units.

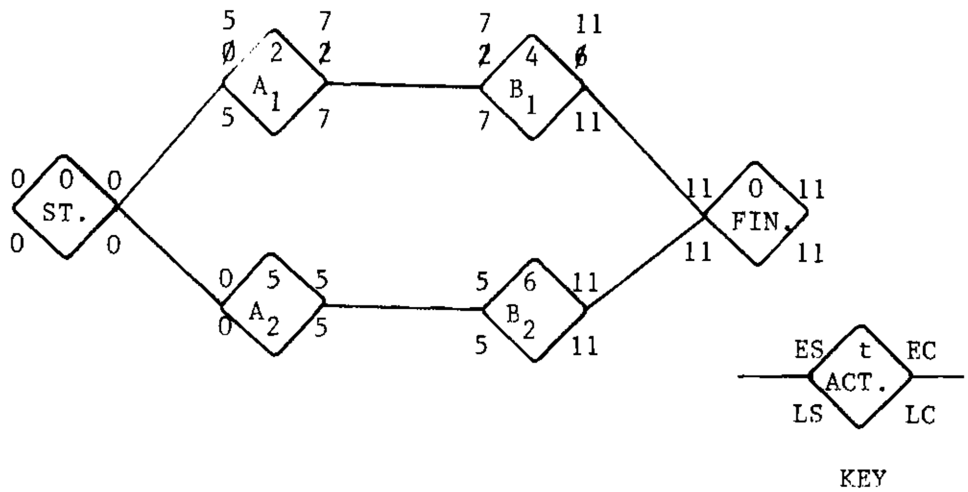


Figure 4. Project Illustrating Suboptimality of RSM

This lack of multi-stage optimality is common to all approaches examined previously. However, RSM does have an advantage over the others in that it gives an optimal single-stage solution.

RAMPS (Resource Allocation and Multi-Project Scheduling System) (21) is an approach to the allocation of limited resources over several projects simultaneously. An explanation (19) of the problem involved and the approach used follows:

> There are hundreds and even thousands of possible schedules for even the simplest RAMPS problem . . . and the number increases exponentially as the number of activities, projects, and resources goes up. For this reason, it is impractical, even with the fastest of modern computers, to attempt to optimize solutions to problems of this kind, i.e., prove mathematically or otherwise that the entire schedule for all projects is the best obtainable under the stated conditions and restric-

tions. The times involved in examining and evaluating all the possible schedules would be prohibitively high.

It is feasible, however, to break the problem into segments, optimize the schedule for each segment, and knot together the schedules for all segments into complete project schedules.

This method is used by RAMPS. The problem is divided into time periods of an hour, day, month, or any predetermined length of time. For each period, RAMPS computes the best combination of scheduled, delayed, and/or interrupted activities according to resource availability, the management objectives, and costs. Optimum work rates are also computed according to the demands of the project completion times.

Other Networking Applications

Eisner (6) describes a generalization of the PERT network approach that allows for the consideration of alternative successors at various stages of a project. The selection of the alternatives to be used cannot be made in advance, for it is the nature of the outcome of the predecessors that dictates which alternative is best. As there is no single fixed plan, there can be no conventional schedule. Eisner's emphasis is directed toward evaluating the uncertainty of alternative networks, each of which contains sets of alternative successors. This evaluation is made in terms of relative entropy. Eisner's work has some aspects which are of value in other types of networking problems, for he introduces the concept of a single network to represent all possible ways in which the project may be carried out.

Burgess and Killebrew (2) utilize the network concept in an algorithm which levels the resource requirements in a repetitive project situation. The situation is more akin to what is thought of as a production process than to a project. However, each unit being produced is sufficiently important to have managerial attention focused on it, not instead of, but in addition to the attention focused on the functional areas involved. The algorithm presented does not assure a minimum variance in resources required by time unit, but does provide a systematic means for improving a given schedule.

CHAPTER III

THE SINGLE NONSIMULTANEOUS SET WITH A SIMULTANEITY MAXIMUM OF ONE

Introduction

The presence of the nonsimultaneity constraint among the activities of a project imposes a requirement which one can be assured of meeting if pseudo-precedence requirements can be added between the activities constituting the nonsimultaneous set. This chapter deals with a nonsimultaneous set in which none of the activities has any true, or technological, precedence requirements imposing partial sequences on the set. If a schedule is based only on the basic network, there might be periods of time in which two or more members of the nonsimultaneous set are scheduled to be in progress concurrently.

If the region consisting of the period of time from the ES of an activity to the LC of that activity is termed the feasible domain of that activity, then activities whose feasible domains are widely separated are not likely to be scheduled to be in progress simultaneously. The initial boundaries of the feasible domains are based on technological precedence only, however, and the imposition of nonsimultaneity constraints on activities earlier in the schedule may cause some or all of these initial boundaries to be shifted to later points in time. Therefore, only if activities are forced to lie on a single precedence path can one be assured that the scheduling process will not violate the nonsimultaneity constraint on the specified set of activities.

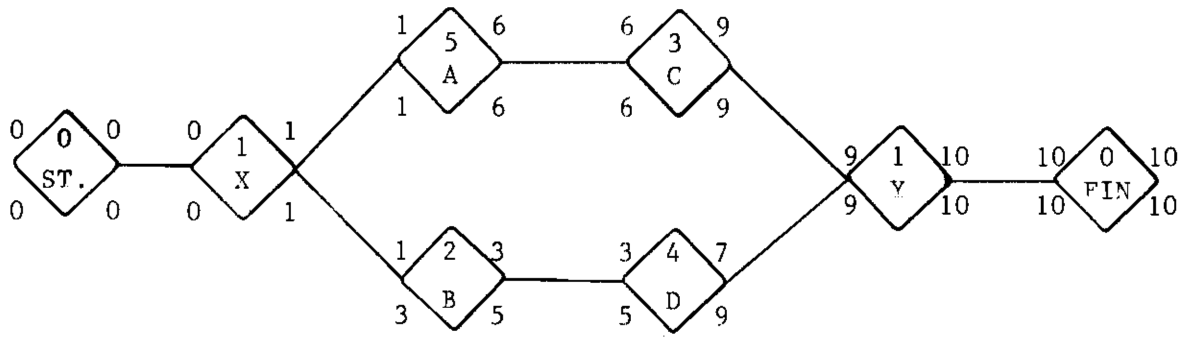
If two activities, neither of which necessarily precedes the other, are nonsimultaneous activities, then adding the requirement that one of them precedes the other will assure nonsimultaneity. If the activities are identified as A and B, then one can require that A >> B or that B >> A. If the only criterion for scheduling is the avoidance of A - B simultaneity, each alternative is equally acceptable. The criterion applied herein, however, is the criterion of minimizing the length of the longest path from START to FINISH passing through any of the activities in the nonsimultaneous set.

If the critical path of the basic network extends through one or more activities from a nonsimultaneous set, the sequencing of the activities in that set is likely to affect the length of the project. If the critical path of the basic network does not extend through any of the activities of the nonsimultaneous set, the sequencing of those activities could still cause some or all of the activities to become critical, thus affecting the project length. If, however, no possible sequence of activities in the nonsimultaneous set could cause any activity in that set to become critical, it is still likely that some sequences have activities whose criticality is greater than the maximum criticality found in other sequences. In general, those sequences with smaller values of minimum slack are more likely to become critical and thus affect the project length. Thus the criterion used has the effect of tending to minimize project length, whether the sequence chosen lies on the critical path or not. The sequence to be selected is the one giving the minimum value of maximum criticality over the activities in the nonsimultaneous set.

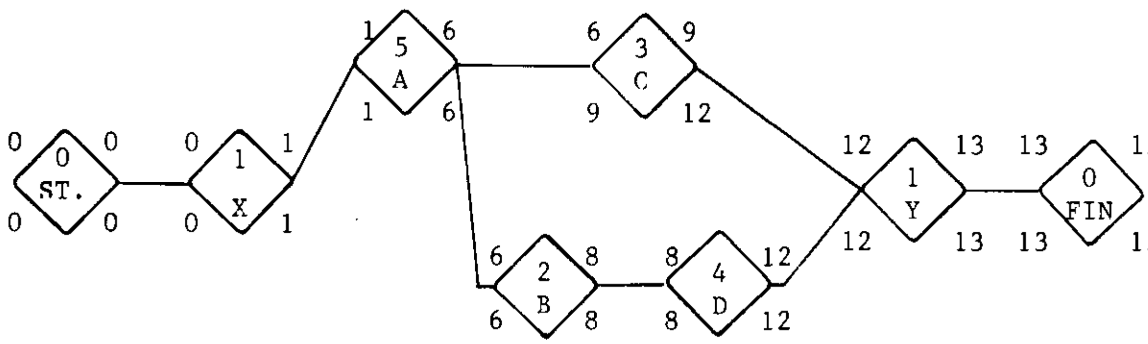
By definition, C(A), the criticality of activity A is dependent upon the maximum EC over the set P_A and the minimum LS over the set S_A. Thus a nonsimultaneous set of activities cannot be optimally sequenced independently of the remainder of the network of which they are a part. Neither can they be optimally sequenced with only a forward pass through the network or only a backward pass through the network.

The complete enumeration approach to the non-simultaneity problem involves the construction of a separate network for every possible sequence of the activities in the nonsimultaneous set. A forward and backward pass through each network would allow the optimum sequence to be identified. If the basic network is small and the number of activities in the nonsimultaneous set is small, this approach is not unreasonable. For example, if in the basic network of Figure 5 (a), page 40, activities A and B form a nonsimultaneous set, then Figure 5 (b) and Figure 5 (c) represent the two alternative ways of using precedence restrictions among the members of the nonsimultaneous set to assure nonsimultaneity.

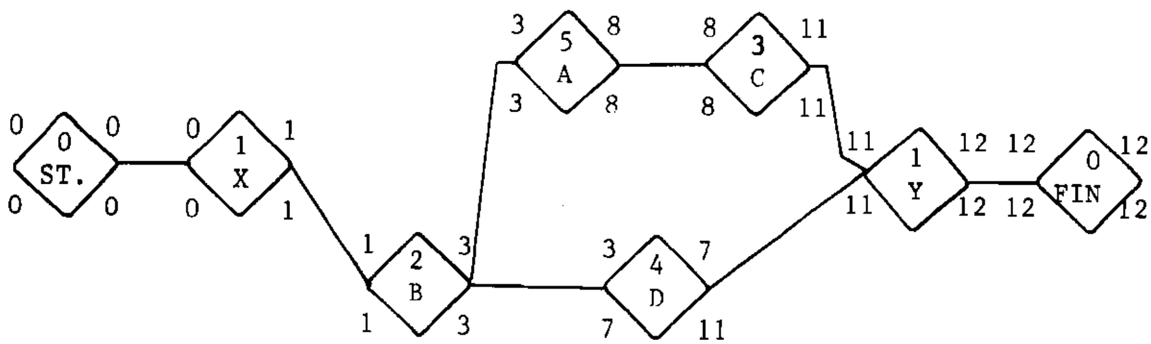
Requiring that B >> A results in a project length which is less than the project length resulting from the requirement that A >> B. It is worthy of note that B should be scheduled ahead of A although A is more critical than B in the basic network. Suppose, however, that the basic network has contained activity E, as shown in Figure 6 (a). Then the choice of sequence would not directly effect the project length.



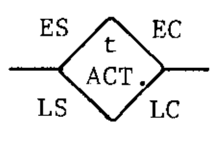
(a) Basic Network



(b) Modified Network, A >> B

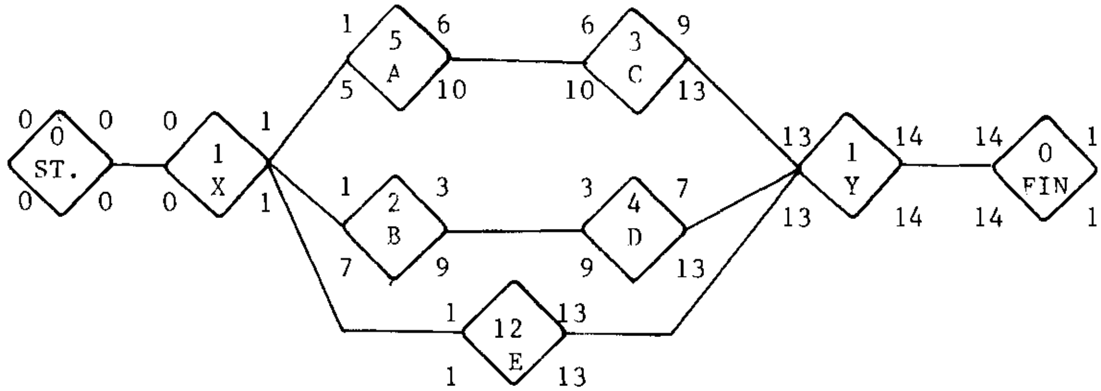


(c) Modified Network, B >> A

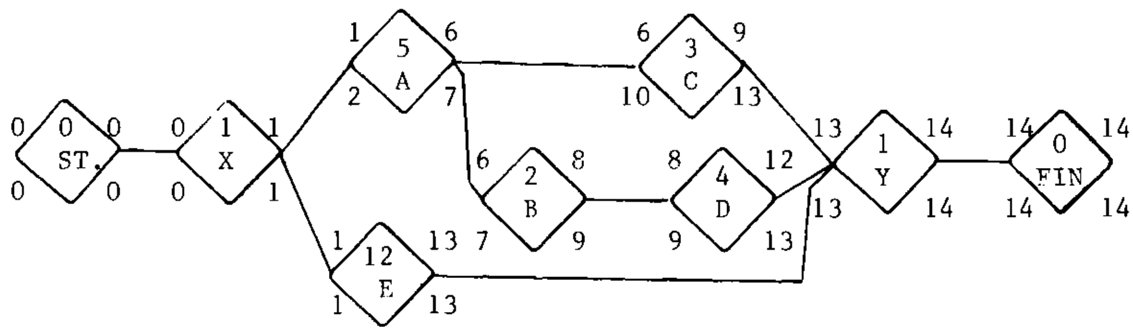


KEY

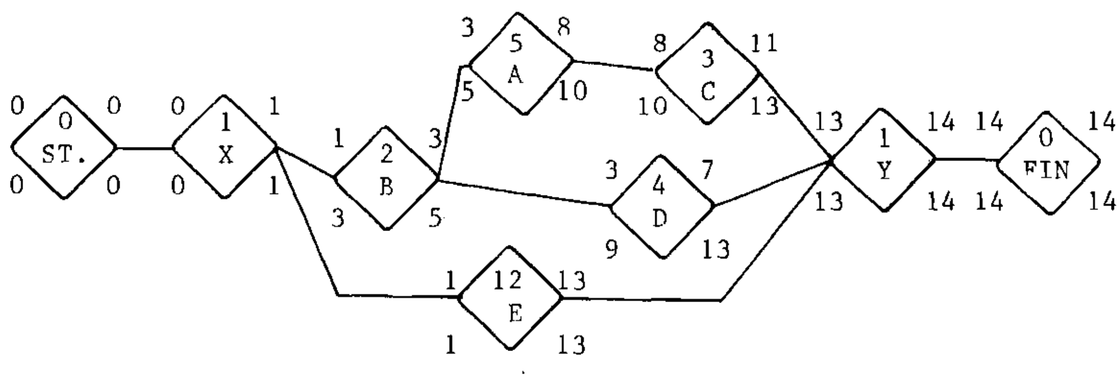
Figure 5. Nonsimultaneity Achieved Through ">>" Relationships on the Critical Path



(a) Basic Network



(b) Modified Network, A >> B



(c) Modified Network, B >> A

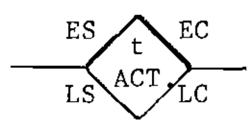


Figure 6. Nonsimultaneity Achieved Through ">>"
Relationships on Noncritical Path

Yet we would still prefer to have B >> A, as in Figure 6 (c), as that sequence allows A and B to slip two time periods relative to the critical path before becoming critical, while letting A >> B, as in Figure 6 (b), gives only one time period of negative criticality to each of the two activities.

The Proposed Approach

General Description

The basic approach developed in this research involves showing all possible sequences of the activities in the nonsimultaneous set in a single network. Then sequences are eliminated by comparing each sequence with the other sequences until the optimal sequence is the only sequence remaining.

Implementation Procedure

The proposed approach is implemented by the following the steps outlined below:

Step One. Identify all members of the nonsimultaneous set.

Step Two. Describe every possible precedence permutation (sequence) which will assure the required nonsimultaneity. Identify each sequence with a number, $j (j = 1, 2, \ldots)$. Assign the sequence number as a subscript of every activity identifier in that sequence.

Step Three. Revise the listing of activities and precedences.

(a) Add the activities and precedence requirements necessary to establish the sequences described in Step Two.

(b) Eliminate the activities composing the nonsimultaneous set.

(c) Make each j-subscripted activity dependent upon every activity which was a predecessor to the non j-subscripted version of the activity in question.

(d) Make each j-subscripted activity a predecessor to each activity which had the non-j-subscripted version of that activity as a predecessor.

(e) Eliminate any redundant precedence relationships created by the above steps. Refer to the network at this stage as an enumerative network.

Step Four. Make a forward pass through the enumerative network and a backward pass from FINISH through the j-subscripted activities. When making network calculations on an enumerative network, some modifications are made in the calculation procedure. If the activity identifiers in the nonsimultaneous set are considered as the set $\{A_i\}$, $i = 1, 2, 3, \ldots, m$, then an activity from a nonsimultaneous set will be identified by an i-j designation in the enumerative network. If an activity depends upon j-subscripted activities, only the $\min_{i,j}$ EC of each subscripted predecessor A_{ij} is considered in determining the ES of the activity in question. If the activity in question also has non-j-subscripted predecessors, the EC of each such predecessor and the $\min_{i,j}$ EC of each subscripted predecessor A_{ij} are compared as in calculations on a basic network to determine the ES of the activity in question.

Step Five. Identify the sequences or sequence j associated with MMC'.

Step Six. If MMC' is associated with a single value of j, retain the sequence associated with that j value. Eliminate the activities and precedence relationships associated with all other sequences. The resulting network will assure that the nonsimultaneity constraint is met with the minimum increase, or tendency for increase, in project length.

Step Seven. If MMC' is associated with more than one value of j, eliminate the activities and precedence relationships associated with the values of j not involved in MMC'. Then in each sequence j remaining, temporarily remove from consideration that value of i which gave MC'.

Step Eight. Repeat Step Five and either Step Six or Step Seven, as applicable, continuing until Step Six is executed or, in Step Seven, the temporary removal from consideration leaves no activity for consideration.

Step Nine. If more than one value of j is still associated with MMC', examine the activities in the set S_{A_i}, the immediate successors to all activities A_i in the basic network. For a given sequence, j^*, from among the remaining sequences j, temporarily replace the ES values for the members of S_{A_i} with the ES values which would result from the selection of sequence j^*. Calculate $\max_{S_{A_i}}$ criticality. Repeat this procedure for each of the remaining sequences. Choose the sequence j associated with $\min_{j} \max_{S_{A_i}}$ criticality. If a unique value for j still is not determined, the remaining sequences are examined by removing from consideration the most critical successor associated with each remaining j and again applying the $\min_{j} \max_{S_{A_i}}$ criticality rule. Continue the elimination of sequences and successors in this manner until either a single sequence remains or all successors are eliminated from consideration.

If more than one sequence remains after all successors have been eliminated from consideration, all the remaining sequences are optimal sequences. Randomly eliminate all but one.

Step Ten. Make a new forward and backward pass through the network. Refer to this network as the modified network. The length of the critical path in the modified network. The length of the critical path in the modified network is the minimum which can be achieved while adding pseudo-precedence requirements to assure nonsimultaneity of the given set of activities.

Example Application

As an example, let the proposed procedure be applied to the network of Figure 6 (a). The basic network has the following tabular description: A_1, A_2 represent A and B, respectively.

Activity	Predecessors	Duration
START	None	0
X	START	1
A_1	X	5
A_2	X	2
C	A_1	3
D	A_2	4
E	X	12
Y	C,D,E	1
FINISH	Y	0

Step One. The nonsimultaneous set consists of A_1 and A_2.

Step Two. Adding $A_1 >> A_2$ will avoid $A_1 - A_2$ simultaneity. Call this sequence $j = 1$. Let $j = 2$ indicate $A_2 >> A_1$.

Step Three. This step leads to the following tabular description:

Activity	Predecessors	Duration
START	None	0
X	START	1
A_{11}	X	5
A_{12}	A_{22}	5
A_{21}	A_{11}	2
A_{22}	X	2
C	A_{11},A_{12}	3
D	A_{21},A_{22}	4
E	X	12
Y	C,D,E	1
FINISH	Y	0

The resulting enumerative network is shown in Figure 7, page 47.

Step Four. The earliest and latest start and completion times shown for the activities in Figure 7 are the result of applying the network calculation procedure described in Step Four of the generalized procedure.

Step Five. MMC' is found from these figures:

Activity A_{ij} i	j	LC'	EC'	Criticality'
1	1	7	6	-1
2	1	9	8	-1
1	2	10	8	-2
2	2	5	3	-2

MMC'

MMC' is associated with j = 2.

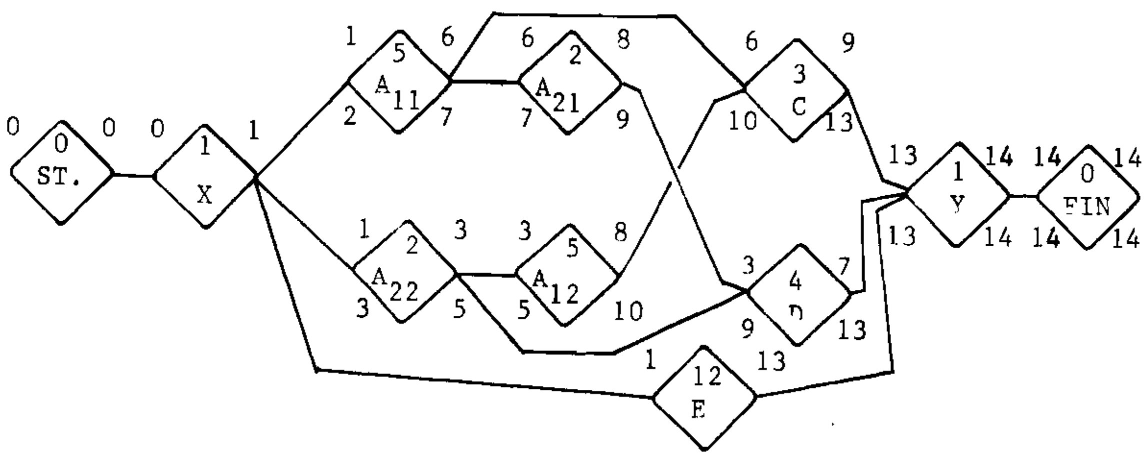


Figure 7. Enumerative Network Based on Figure 6 (a) with A_1 and A_2 in the Nonsimultaneous Set

Step Six. This step, rather than Step Seven, applies and results in the elimination of the sequence j = 1 and its associated precedence relationships. The modified network of Figure 6 (c) results.

Steps Seven, Eight and Nine. These steps do not apply in this case.

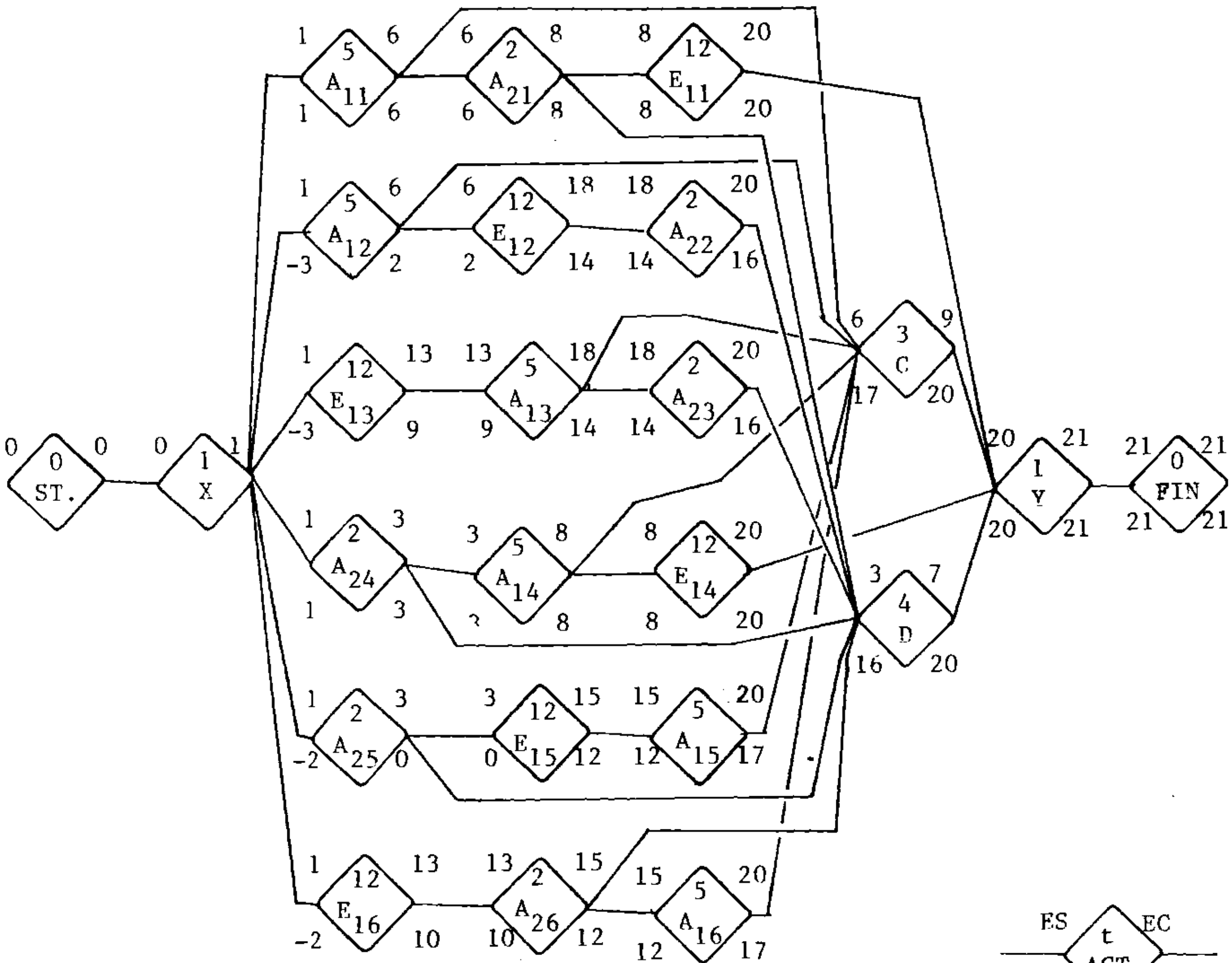
Step Ten. The earliest and latest start and completion times shown with the network of Figure 6 (c) are the times which would result from a forward and backward pass through the modified network.

If, in the preceding example, activities A_1, A_2, and E_1 form a set such that only one of them may be in progress at any one time, the suggested procedure leads to the enumerative network of Figure, shown on the following page. MMC' is associated with j = 1 and j = 4. Application of Steps Seven, Eight, and Nine shows that j = 4 is the optimal sequence.

Proof of Optimality of Foregoing Procedure

Generalized Network Representation of Set Relationships

In Figure 9, shown on page 50, the A_{ij} are alternative sequences for assuring the nonsimultaneity of a nonsimultaneous set, $\{A_i\}$. The X's include START and are predecessors, though not necessarily immediate predecessors, to one or more of the members of i. The Y's include FINISH and are successors, though not necessarily immediate successors, to one or more of the members of i. Any precedence relationship not covered by the above is possible for the activities in the set Z. This implies that the Z's are not precedence-related to any member of i.



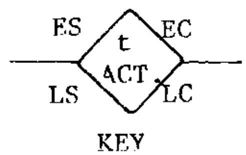


Figure 8. Enumerative Network Based on Figure 6(a) with A_1, A_2 and E Composing the Nonsimultaneous Set

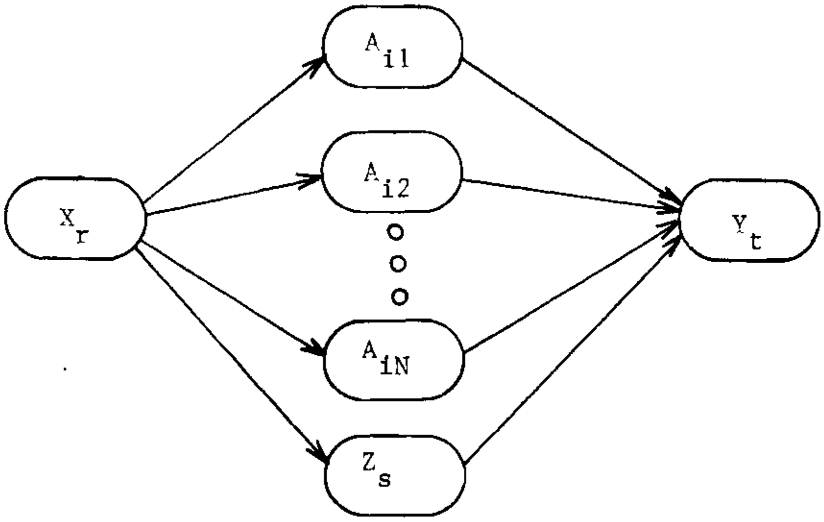


Figure 9. Generalized Representation of a Single Set Enumerative Network

In Figure 9, a precedence arrow between sets indicates dependence of one or more activities in a set on one or more activities in another set.

Before using this generalized representation to prove the optimality of the foregoing procedure, it is desirable to present and prove two theorems which are of a general nature. These theorems should prove useful in a variety of problems not discussed in this paper.

Theorems on Transfer of Criticality

Theorem One: No activity can be more critical than the most critical of its immediate predecessors.

Proof: Let the activity in question be activity A. The set of immediate predecessors, P_A, contains at least one member, activity B, such that EC(B) = ES(A), and no member, activity X, such that EC(X) > ES(A). These two conditions hold by virtue of the definition of ES, given in Equation (2). Also, by virtue of the definition of LC, given in Equation (4), it can be shown that LC(B) ≤ LS(A).

The criticality of activity B, C(B), is established through Equation (5) and (7) as C(B) = EC(B) - LC(B). Similarly, C(A) = ES(A) - LS(A). As EC(B) = ES(A) and LC(B) ≤ LS(A), C(A) ≤ EC(B) - LC(B) = C(B). Thus the criticality of A is equal to or less than the criticality of B, where B is any activity whose earliest completion is the same as the earliest start of A.

There may be other members of P_A, such as activity D, such that EC(D) < ES(A) = EC(B). The condition C(D) ≥ C(B) implies that C(D) ≥ C(A). The most critical activity in P_A will be an activity B or an activity D for which C(D) > C(B). Thus C(A) is equal to or less than the criticality of the most critical of its predecessors and Theorem 1 is proven.

Theorem Two: No activity can be more critical than the most critical of its immediate successors.

Proof: Let the activity in question be activity A. The set of immediate successors, S_A, contains at least one activity, say activity E, such that LC(A) = LS(E) and no member, X, such that LS(X) < LC(A). These conditions hold by virtue of the definition of LC (Equation (4)). Also, by definition of ES (Equation (2)), ES(E) ≥ EC(A).

The criticality of E is given by ES(E) - LS(E) = C(E). The criticality of A is given by C(A) = EC(A) - LC(A). Using the fact that LC(A) = LS(E), C(A) = EC(A) - LS(E). Using the fact that ES(E) ≥ EC(A),

$C(A) \leq ES(E) - LS(E) = C(E)$.

S_A may contain some activity F such that $C(F) \geq C(E)$ and $LS(F) > LS(E)$. Therefore, $C(F) > C(A)$. The most critical activity in S_A will be an activity E or an activity F for which $C(F) \geq C(E)$. Thus C(A) is equal to or less than the criticality of the most critical of its immediate successors. Theorem Two is thus proven.

Proof of Optimality of Nonsimultaneity Procedure

The EC of a project, based on a forward pass using nonsimultaneity procedures, will be associated with one or more Y_t. Using any LC for the project and making a backward pass through the A_{ij} and the Z_s, some A_{ij}, say A_{i2}, will give MMC' over the A_{ij}. By Theorem 2, deletion of all A_{ij} except A_{i2} leaves the maximum criticality among the Y_t equal to or greater than $MC(A_{i2})$.

If $\max_t C(Y_t) = \max_i C(A_{i2})$, the substitution of some other sequence, say A_{i3}, for A_{i2} would cause the $\max_t C(Y_t)$ to be equal to or greater than the $\max_i C(A_{i3})$. But $\max_i C(A_{i3}) \geq \max_i C(A_{i2})$. Thus the substitution of A_{i3} for A_{i2} would make the maximum criticality among the Y_t equal to or greater than the maximum criticality among the Y_t before the substitution. As the LC for the Y_t are not affected by the choice of sequences, A_{ij}, the criticality of the Y_t can be affected only by changing the EC of the Y_t. The Y_t having maximum criticality are on the critical path. Thus the maximum value of EC among the most critical Y_t minus the minimum value of ES among the X_r gives the minimum project length. Therefore, the substitution of A_{i3} for A_{i2} either will not effect the minimum project length or will increase the minimum project length.

If the $\max_t C(Y_t) > \max_i C(A_{i2})$, at least one path through the X_r and Z_s is critical. As such paths contain no members of A_{i2}, the substitution of A_{i3} for A_{i2} cannot alter the length of such paths. Therefore, the substitution of some other A_{ij} for A_{i2} cannot shorten the project length. If, however, $\max_i C(A_{i3}) > \max_t C(Y_t)$ when A_{i2} was the only sequence considered, the substitution of A_{i3} for A_{i2} will increase the project length. If $\max_i C(A_{i3}) > \max_i C(A_{i2})$, but $\max_i C(A_{i3}) \leq \max_t A_{i2}$ was the only sequence considered, substitution of A_{i3} for A_{i2} will not directly affect the minimum project length. However, such substitution will decrease the minimum slippage required among the A_i to increase the minimum project length.

These findings lead to the conclusion that the sequence A_{ij} having $\min_j \max_i$ criticality is the optimal sequence to assure nonsimultaneity of the A_i for a project whose network model can be generally described as in Figure 9.

Computational Magnitude of Proposed Approach as Compared with Complete Enumeration of all Possible Networks

As the number of activities in the nonsimultaneous set is increased from two to three, the number of alternative networks increases from 2! = 2 to 3! = 6. In general, if only one of N activities can be in progress at any one time, and if there are T activities in the basic network, full enumeration of alternative networks which assure nonsimultaneity involves T(N!) activities. The single enumerative network approach involves T - N + N(N!) activities. In number of activities to be dealt with, the single enumerative network approach involves a saving of T(N!) - T + N - N(N!) = (T - N)(N! - 1). Table 1 shows the difference between the two approaches for several values of T and N, with A representing full enumeration of all networks and B representing the single enumerative network approach.

Table 1. Number of Activities to Be Dealt with Using the Complete Enumeration Approach (A) vs. the Enumerative Network Approach (B)

			T					
			10	20	50	100	500	1,000
N	2	A	20	40	100	200	1,000	2,000
		B	12	22	52	102	502	1,002
	3	A	60	120	300	600	3,000	6,000
		B	25	35	65	115	515	1,015
	4	A	240	480	1,200	2,400	12,000	24,000
		B	102	112	142	192	592	1,092
	5	A	1,200	2,400	6,000	12,000	60,000	120,000
		B	605	615	645	695	1,095	1,595
	7	A	50,400	100,800	252,000	504,000	2,502,000	5,040,000
		B	35,283	35,293	35,323	35,373	35,773	36,273

CHAPTER IV

MULTIPLE NONSIMULTANEOUS SETS WHEN THE SIMULTANEITY MAXIMUM IS EQUAL TO ONE

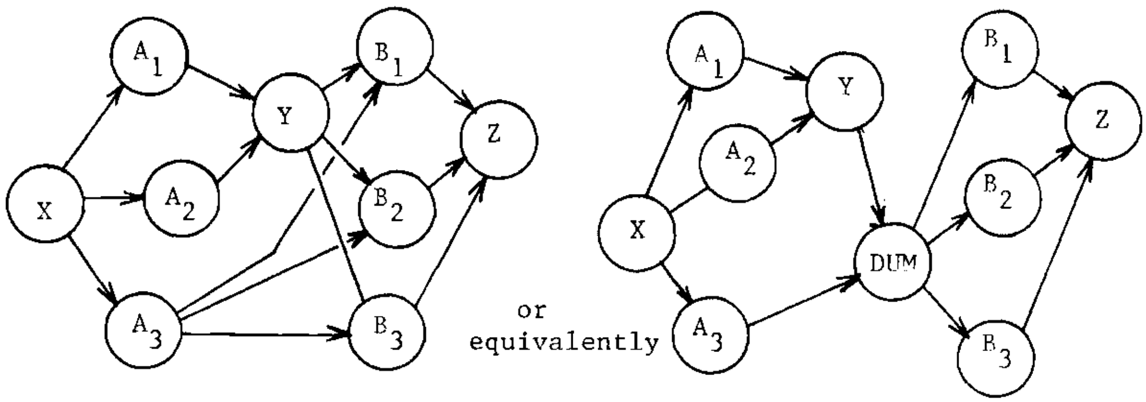
Independent Sets

A network may contain more than one nonsimultaneous set. Two such sets will be referred to as independent if the selection of the sequence giving MMC' for one set is independent of the sequence selected for the other set. Two sets, $\{A_i\}$ and $\{B_i\}$, will be independent if:

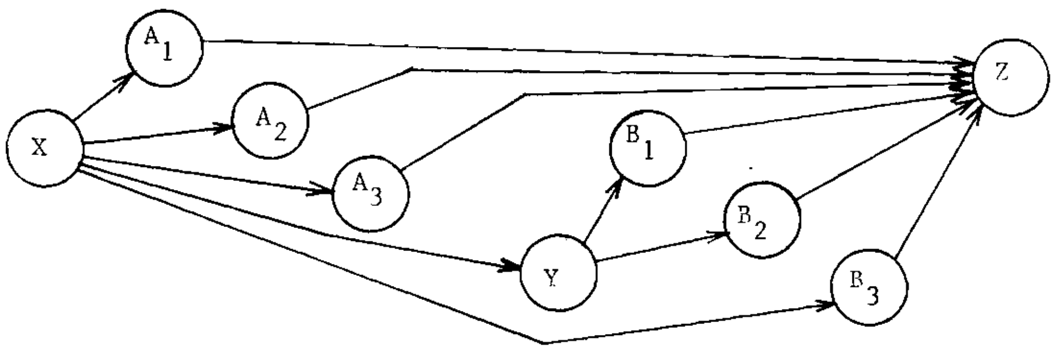
(1) (a) all activities in either or both of the sets have identical precedence relationships between themselves and activities of the other set such that the network could be drawn with all precedence connections between sets channelled through a single dummy node; or, (b) no path of precedence relationships extends through both $\{A_i\}$ and $\{B_i\}$; and

(2) there is no third set, $\{C_i\}$, such that dependency exists between $\{A_i\}$ and $\{C_i\}$ and between $\{B_i\}$ and $\{C_i\}$. Otherwise, dependency will be said to exist between $\{A_i\}$ and $\{B_i\}$. Figure 10 shows examples of independence and dependence between sets A_i and B_i.

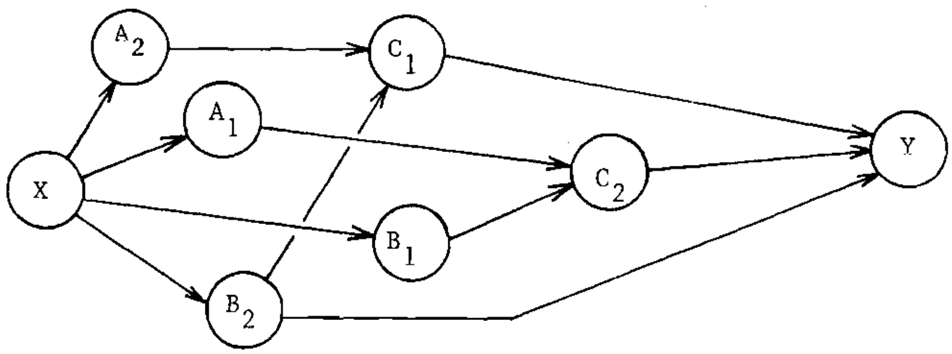
Figure 11 (a), page 57, shows the modified network for the basic network shown in Figure 10 (c). This is the optimum sequencing (found by complete enumeration) to meet nonsimultaneity constraints, given the durations indicated.



(a) Conditions 1(a) and 2 Met

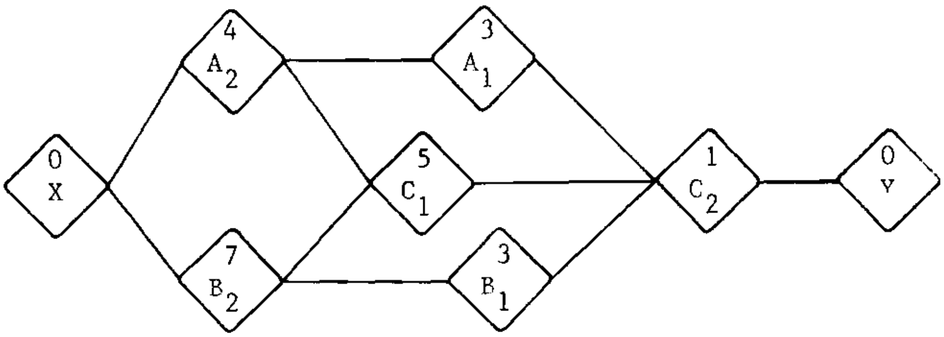


(b) Conditions 1(b) and 2 Met

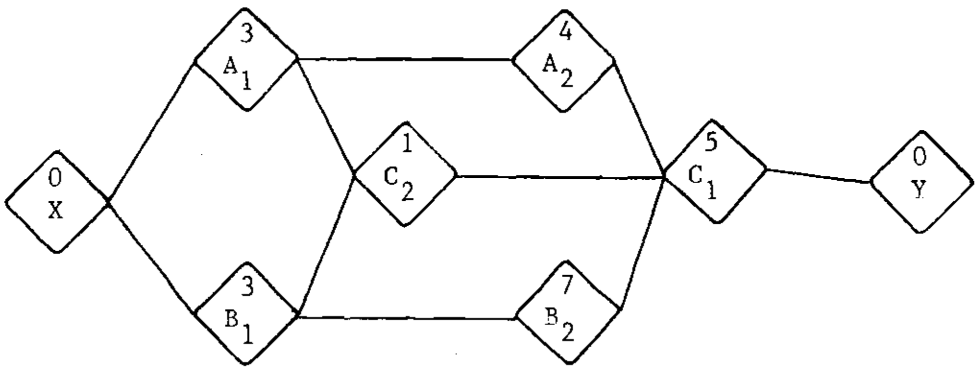


(c) Conditions 1(a) and 1(b) Met for A and B, not Met for A and C or B and C, Condition 2 not Met

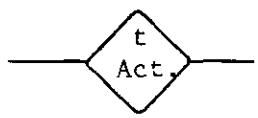
Figure 10. Examples of Independent and Dependent Nonsimultaneous Sets



(a) Optimal Sequences if Choice is Unrestricted



(b) Optimal Sequences for A_i, C_i Given $B_1 >> B_2$



KEY

Figure 11. Modified Networks for the Basic Network on Figure 10

If the sequence in nonsimultaneous set $\{B_i\}$ were reversed, the network of Figure 11 (b) would represent the best sequencing of $\{A_i\}$ and $\{C_i\}$. This example indicates that the selection of a sequence for a given set may influence the selection in some otherwise independent set if the two sets are both dependent on the same third set, i.e., condition (2) above is not met.

The optimal selection of sequences in networks containing multiple nonsimultaneous sets, all of which are independent, is conducted in a manner quite like that presented in Chapter III. The only difference is that the initial backward pass does not terminate until all nonsimultaneous sets have been included. As soon as all latest times are calculated for the members of a nonsimultaneous set and before any other latest times are determined from these, the optimum sequence is chosen and all others eliminated from the enumerative network. The backward pass then continues, if necessary, to make it possible to select the optimum sequence in some other nonsimultaneous set. When optimum sequences have been selected for all nonsimultaneous sets, a new forward and backward pass can be made to allow for the determination of the true criticality of all activities in the network.

Comparing the single enumerative network approach with full enumeration of all possible modified networks, the number of activities to be dealt with by using the former was shown to be much smaller when only a single nonsimultaneous set is involved. The advantage of the single enumerative network approach is even more striking when multiple nonsimultaneous sets are involved. As an example, consider a 200-activity network containing three independent nonsimultaneous sets of three activities each. Full enumeration of all possible modified networks would involve one calculation of ES, EC, LC, and LS for each of 200 $(3!)^3$ = 43,200 activities. The single enumerative network approach would require (200 - 9) + 3(3!)3 = 245 activities, involving a minimum of one calculation of ES, EC, LC, and LS for each of 191 activities and two such calculations for each of 54 activities. The maximum number of such calculations would be two for each of 245 activities. The reduction factor in calculation time would be at worst 2(245)/43,200 = 0.011.

Dependent Sets

Introduction

If a network contains multiple nonsimultaneous sets, it may not be possible or feasible to prove that the sets are independent. Therefore, an approach which handles both independent and dependent nonsimultaneous sets in the same network would be much more generally useful. The following development shows how the procedure developed for independent nonsimultaneous sets gives optimal answers for networks consisting of a series of dependent sets. A modification is then made to further generalize the procedure. This generalization allows the use of the procedure on networks containing some nonsimultaneous sets through which the critical path of the modified network will pass and other sets, dependent on those containing the critical path, which will not contain any critical activities in the modified network.

A Theorem on EC' - ES' Consistency between Sequences of Adjacent Nonsimultaneous Sets

The question which quickly arises when dealing with dependent nonsimultaneous sets is whether or not the ES' of a given sequence can all be obtained when only one sequence of a preceding nonsimultaneous set is in effect. Theorem Three states that the answer is affirmative.

Theorem Three: Given that two nonsimultaneous sets, A_i and B_i, exist such that one or more activities in B_i depend upon one or more activities in A_i, there is at least one sequence, A_{ij}, which, when used as the only sequence of the A_i, will allow $ES(B_{iJ}) = ES'(B_{iJ})$ for all members of a chosen sequence, B_{iJ}.

Proof: Let B_{iJ} be represented as in Figure 12.

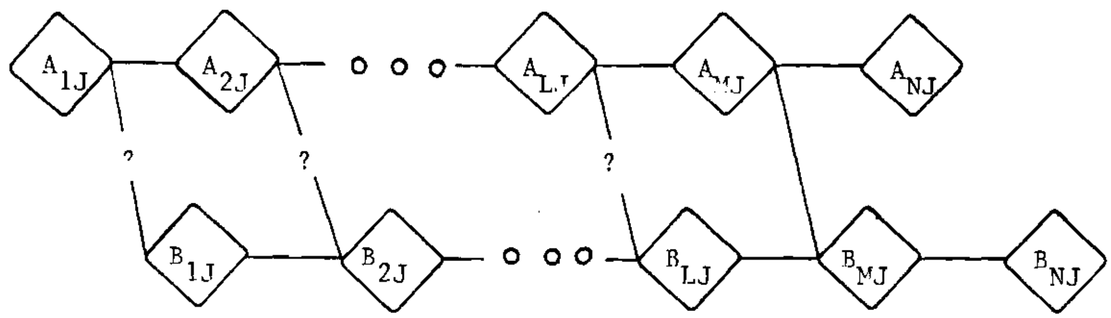


Figure 12. Relationship between Sequences from Adjacent Nonsimultaneous Sets

Determine the maximum value iJ for which some activity A_{ij} exists such that $A_{ij} >> B_{iJ}$. Let A_{iJ} be the sequence which gives minimum $EC'(A_{ij})$ for $A_{ij} >> B_{iJ}$. Let the A_{iJ} and B_{iJ} so related be termed A_{MJ} and B_{MJ}. Figure 12 illustrates this relationship. Arrows broken by question marks indicate possible, but not necessary, precedence relationships.

The $ES'(B_{iJ})$ for activities B_{iJ} following B_{MJ} will be unchanged if $ES'(B_{MJ})$ is unchanged. $ES'(B_{MJ})$ will be unchanged if sequence A_{iJ} is chosen and $ES'(B_{LJ})$ is unchanged.

If $A_{ij} >> B_{iJ}$, these A_{ij} will be predecessors, A_{iJ}, of A_{MJ}. Otherwise, the redundancy examination would have disallowed $A_{MJ} >> B_{MJ}$. Revising the sequence of predecessors to A_{MJ} will not affect the $ES'(A_{MJ})$ and will allow the attainment of the $ES'(B_{LJ})$ if $EC'(B_{2J})$ is unchanged. If B_{2J} is the only immediate predecessor of B_{LJ}, one needs only assure that $EC'(B_{2J})$ will not change to assure that $ES'(B_{LJ})$ will not change.

If $A_{ij} >> B_{2J}$, these A_{ij} will be predecessors, A_{iJ}, of those activities in sequence A_{iJ} found to be predecessors of any B_{iJ}, $i > 2$. These A_{ij} can be sequenced to attain the minimum $EC'(A_{ij})$ for $A_{ij} >> B_{2J}$. This sequencing will not necessitate further resequencing of the activities A_{iJ} found to be predecessors to B_{iJ}, $i > 2$. Thus, if $EC'(B_{1J})$ is unchanged, $ES'(B_{2J})$ will be unchanged.

If $A_{ij} >> B_{1J}$, these A_{ij} will be predecessors, A_{iJ}, of those activities in sequence A_{iJ} found to be predecessors of any B_{iJ}, $i > 1$. These A_{iJ} can be sequenced to attain the minimum $EC'(A_{ij})$ for $A_{ij} >> B_{1J}$.

This sequencing will not necessitate further resequencing of the activities A_{iJ} found to be predecessors to B_{iJ}, $1 > 1$. If B_{1J} depends only on some A_{iJ} or on some activity not a member of set A_i or both, its ES' will be unchanged.

Thus, a sequence of A_{ij} can be chosen to give $ES(B_{iJ}) = ES'(B_{iJ})$ for all i. Furthermore, this statement holds for any sequence, B_{ij}, since nothing in the proof restricts the selection of sequence B_{iJ}. Therefore, for two adjacent nonsimultaneous sets, A_i and B_i, there is a consistent (in terms of $EC' \leqq ES'$) A_{ij} - B_{ij} sequence for every sequence B_{ij}. The theorem is thus proven.

If a third dependent nonsimultaneous set, C_i, is dependent upon B_i, one can consider the consistent A_{ij} - B_{ij} sequences as the sequences A_{ij} in the proof and treat the new sequences C_{ij} as B_{ij} in the proof. Using this approach, one can show that a consistent A_{ij} - B_{ij} - C_{ij} sequence exists for every sequence C_{ij}. This reasoning allows the extension of the theorem to a series of dependent nonsimultaneous sets of any length.

Proof of Optimality of Multi-Set Procedure when Applied to Dependent Sets

Figure 13, page 63, is used to represent an enumerative network developed from a basic network in which one or more activities in each set depend upon one or more activities in the preceding nonsimultaneous set. After a forward pass is made using nonsimultaneity rules, a backward pass is made through set Z_i. The determination of $MMC'(Z_{ij})$ is then made. Let MMC' be associated with $j = J$ in each set. Therefore, $MMC'(Z_{ij}) = MC'(Z_{iJ})$. The other sequences in set Z are deleted and the backward pass continued through set Y.

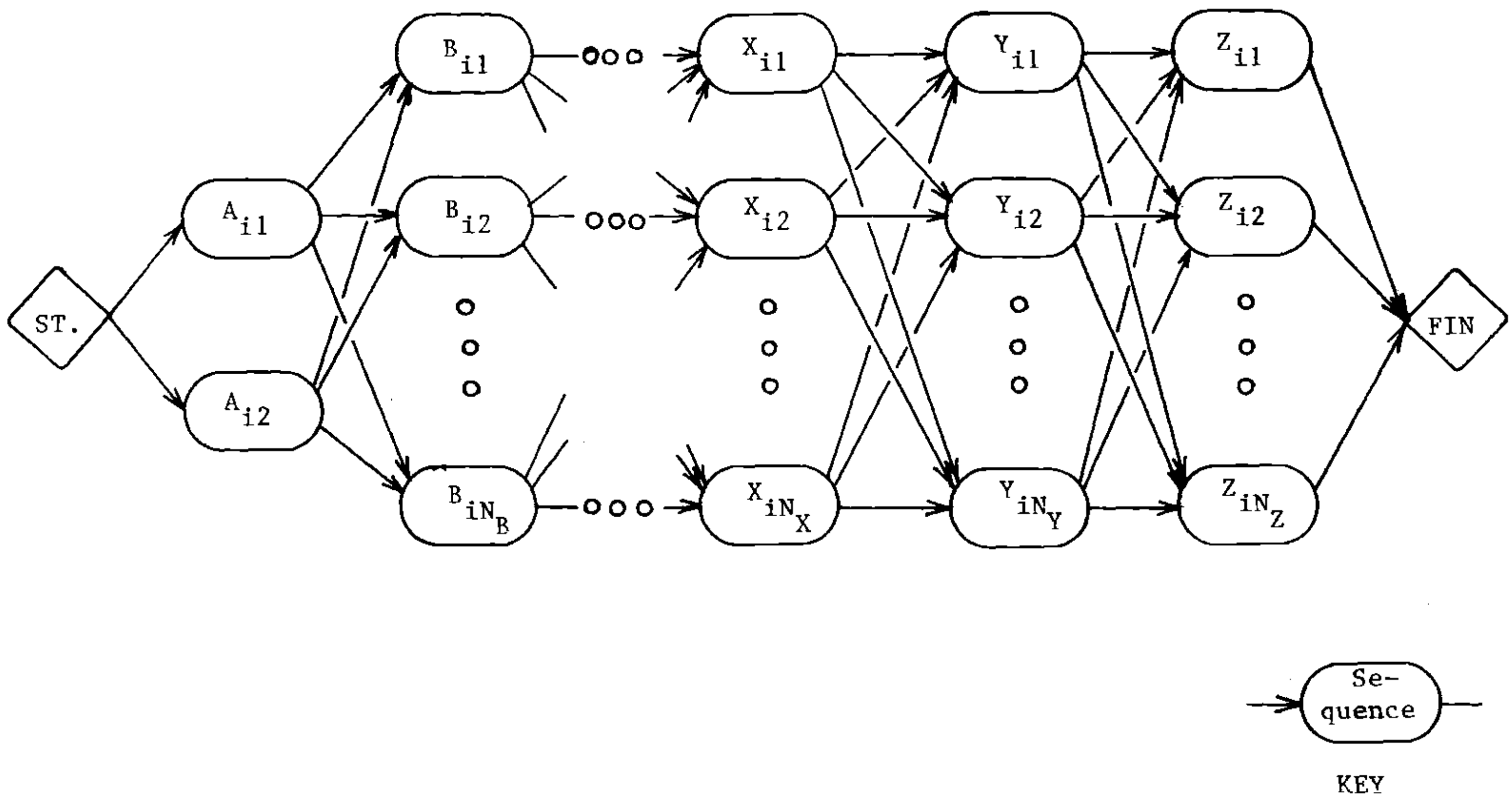


Figure 13. A Generalized Representation of an Enumerative Network of Dependent Nonsimultaneous Sets

$MMC'(Y_{ij})$ is determined and all sequences Y_{ij} other than Y_{iJ} are deleted.

Now assume for a moment that all sequences in set Z are restored. Will MMC after a new forward pass from sequence Y_{iJ} still be associated with Z_{iJ}? The use of sequence Y_{iJ} only in calculating $ES(Z_{ij})$ will cause $ES(Z_{ij}) \geqq ES'(Z_{ij})$. The original forward pass assigned the minimum ES' possible with any sequence of the Y_i. The $LC'(Z_{ij})$ will be unchanged by the change in forward pass procedure. Therefore, $C(Z_{ij})$ based on the second forward pass is equal to or greater than $C'(Z_{ij})$. But by Theorem Three, $ES(Z_{iJ}) = ES'(Z_{iJ})$; therefore, $C(Z_{iJ}) = C'(Z_{iJ})$ and, consequently, $MC(Z_{iJ}) = MC'(Z_{iJ})$. $MC(Z_{ij}) \geqq MC'(Z_{ij})$ for all $j \neq J$. Therefore, $MMC(Z_{ij}) = MMC'(Z_{ij}) = MC'(Z_{iJ})$.

It has previously been shown that for the single nonsimultaneous set the MMC' sequence is optimal. This finding is now used to state that if sequence Y_{iJ} is optimal, then sequence Z_{iJ} is optimal.

Assume now that the backward pass, using Y_{iJ} and Z_{iJ} as the only sequences of their respective nonsimultaneous sets, is carried through the X_{ij}. The MMC' will be assumed to be associated with X_{iJ}. Assume for the moment that all sequences Y_{ij} are restored to the network. Will Y_{iJ} still give MMC' after X_{iJ} is selected as the sequence of X_{ij} to use and a new forward pass is made from the start of X_{iJ}?

The $LS'(Y_{ij})$ will remain constant regardless of the sequence chosen for the X_{ij}. The $ES'(Y_{ij})$ cannot decrease but may increase. This causes $C(Y_{ij})$ using only X_{iJ} to be equal to or greater than $C'(Y_{ij})$ obtained using all sequences of X_{ij}. But as has been previously shown for Z_{iJ} and Y_{iJ}, $C(Y_{iJ}) = C'(Y_{iJ})$. Therefore, $MMC(Y_{ij}) = MMC'(Y_{ij}) = MC'(Y_{iJ})$. It follows that if sequence X_{iJ} is optimal, sequence Y_{iJ} is optimal and thus sequence Z_{iJ} is optimal.

This process can be continued until one reaches the first nonsimultaneous set in the network. At this point, one would know that if A_{iJ} is the optimal sequence among the A_{ij}, then B_{iJ}, C_{iJ}, ..., Z_{iJ} are optimal. Let A_i consist of two dummy activities with zero duration. Thus, both sequence A_{i1} and sequence A_{i2} are optimal. Therefore, the chain of succeeding sequences which was optimal if A_{iJ} was optimal is necessarily optimal.

It thus has been proven that the nonsimultaneity procedure developed for multiple independent sets also is optimal for dependent sets. This holds under the condition that the critical path pass through each set in the enumerative network. One logically would ask at this point how one might know whether or not a specific nonsimultaneity problem would result in such an enumerative network. It is unrealistic to attempt to assure that this condition will be met. The approach taken herein is to further modify the nonsimultaneity procedures in such a manner as to cope with this situation automatically if it occurs.

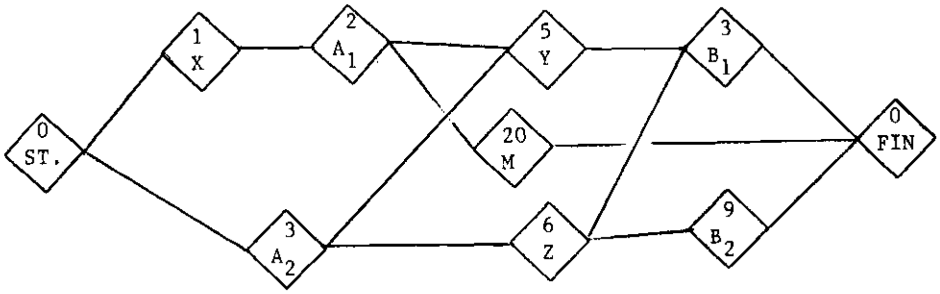
Further Modification to Deal with Networks in which Some Sequences Will Not Be on the Critical Path of the Enumerative Network

When two dependent nonsimultaneous sets are in a network, the MMC' calculated for the second is based on the assumption of free choice of sequences in the first set. When these two sets are the only activities in the network, their MMC' values are the same. This is true because Theorem Three indicates that their earliest times will be consistent and the nonsimultaneity procedures assure that their latest times will be consistent.

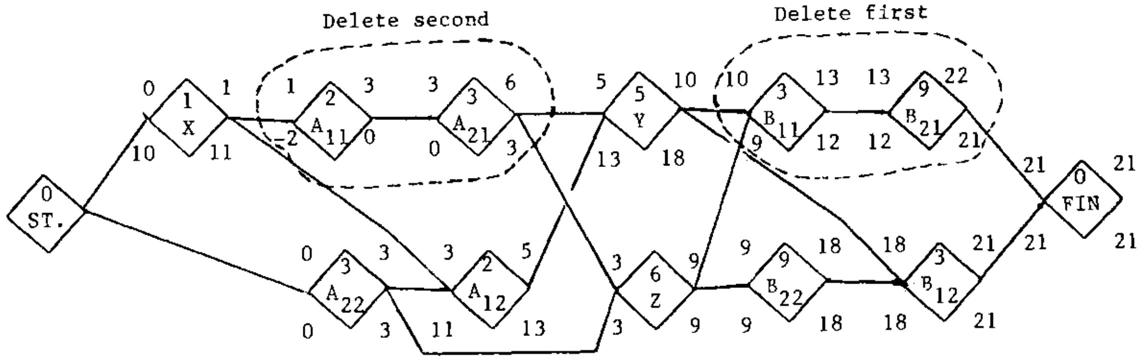
If, however, an additional activity is added to the network and made dependent upon some activity in the first set, it is possible that MMC' for the first set will be larger than MMC' for the second set. Figure 14, on the following page, illustrates this situation.

Figure 14 (b) shows the results of applying the multi-set nonsimultaneity procedure to the basic network of Figue 14 (a), with activity M deleted. The forward pass is made by choosing first at each step the minimum ES' among immediate predecessors from the same nonsimultaneous set. This value is then compared with ES' values from other immediate predecessors. The backward pass is made through the sequences B_{ij} and the MMC' is found to be associated with B_{i2}. B_{i1} is deleted and the backward pass continued through the sequences A_{ij}. $MMC'(A_{ij}) = MC'(A_{22}) = 0$, so sequence A_{i1} is deleted. New forward and backward passes result in no changes except to add latest times for activity X and the initial dummy.

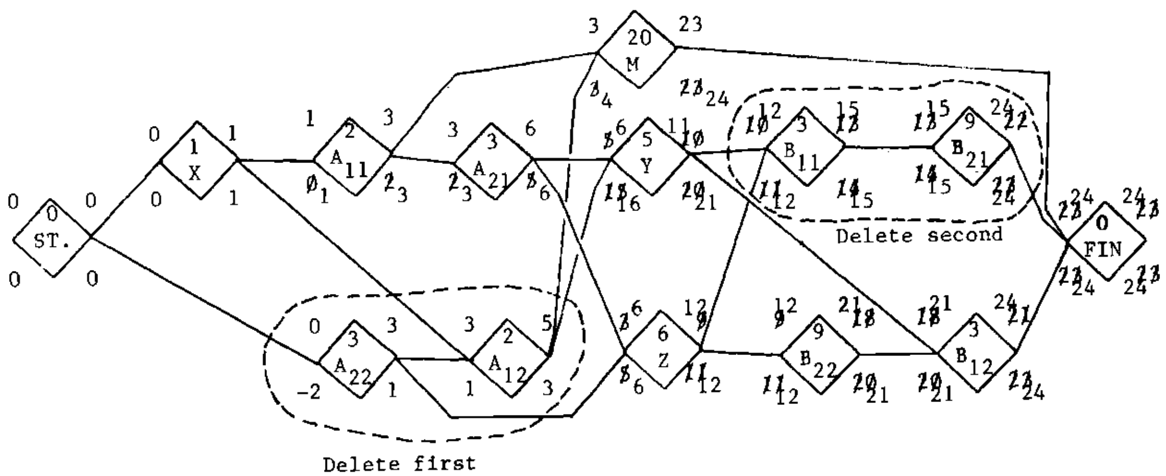
Figure 14 (c) shows the changes created by the inclusion of activity M in the enumerative network. Note first that the EC' of the project is not determined by a sequence B_{ij}. Where two figures appear in the ES, EC, LS, and LC positions, the slashed figures are the results of the first forward and backward pass.



(a) Basic Network with Nonsimultaneous Sets A_i and B_i



(b) Enumerative Network with Sequences to be Deleted During Backward Pass. Activity M Omitted



(c) Enumerative Network with Sequences to be Deleted during Process. Activity M Included

Figure 14. Nonsimultaneity Problem when Enumerative Network Has Critical Path Not Passing through Sequences of One Set

On the backward pass in Figure 14 (c), no sequence B_{ij} is eliminated, but only the LC' values from the MMC' sequence, B_{i2}, are used. Note that $MMC'(B_{ij}) = 21 - 23 = -2$. Continuation of the backward pass through the A_{ij} leads to a determination that $MMC'(A_{ij}) = MC'(A_{i1}) = 6 - 5 = 1$. The fact that sequence A_{i2} has a higher MC' than sequence A_{i1} is directly due to the effect of activity M.

As set A_i is the last set the backward pass will reach, the sequence A_{i2} is deleted and a new forward pass is made. The new forward pass shows that EC for the project is now increased to 24 because the sequence A_{i1} does not allow the shortest $A_{ij} - B_{ij}$ sequence. If the sequence A_{i2} were utilized, however, EC(M) would be 25.

A new backward pass through the sequences B_{ij} leads to the determination that $MC'(B_{i1}) = MC'(B_{i2})$. Elimination of the most critical activity in each set still leaves them tied for MC'. Looking at predecessors to the sequences B_{ij}, one sees that $C'(Z) = 0$ with either sequence. However, $C'(Y) = 11 - 12 = -1$ with sequence B_{i1} in effect and only $11 - 21 = -10$ with sequence B_{i2} in effect. On this basis, sequence B_{i1} is deleted.

The backward pass is now completed through the network. The optimal selection of sequences has been achieved.

The nonsimultaneity procedure used in Figure 14 (c) is applicable to all nonsimultaneity conditions previously considered. The previous solution procedures are in reality subsets of this general procedure. The general procedure is outlined in the following section.

A Generalized Procedure for Sequencing Nonsimultaneous Activities

The following procedure is applicable to networks having one or more independent or dependent nonsimultaneous sets, where each set has a simultaneity maximum of one:

Step One. Construct an enumerative network as explained on pages 42 and 43.

Step Two. Make a forward pass, using the procedure given on page 43.

Step Three. Make a backward pass through the first nonsimultaneous sets encountered. Calculate MMC' over the sequences in each set. Use only the MMC' sequence in continuing the backward pass through the next nonsimultaneous sets. Repeat this process until the forward pass has been extended through all nonsimultaneous sets.

Step Four. Among the initial nonsimultaneous sets, select the set having the maximum value of MMC'. In that set, delete all sequences other than the MMC' sequence.

Step Five. Repeat Steps Two through Four until a sequence has been selected for each nonsimultaneous set. Complete the backward pass from the last sequence selected. The resulting modified network contains the optimal combination of sequences.

The above procedure requires at worst one complete forward and backward pass for each nonsimultaneous set in the network. For a basic network containing 200 activities, including three nonsimultaneous sets of three activities each, the enumerative network would contain 245 activities. The maximum number of activity time calculations would be

3(245) = 735. Full enumeration of all possible modified networks would involve 43,200 activity time calculations.

CHAPTER V

THE NONSIMULTANEITY CONSTRAINT WHEN THE MAXIMUM SIMULTANEITY IS GREATER THAN ONE

Introduction

The most common nonsimultaneity constraint allows only one activity of a set to be in progress at any one time. The number of possible sequences to assure adherence to the constraint is N!, where N = number of activities in the nonsimultaneous set. The N! also gives an indication of the manner in which the sequences may be developed. Each activity will appear in the first position in some sequences. With a given activity in the first position, each of the remaining activities will appear once in the second position, and so on through the last position. When two or more of the N activities are allowed to be in progress simultaneously; however, the procedure requires revision.

Limited Simultaneity Allowed among the Activities of a Nonsimultaneous Set

Designing Alternative Sequences

Let N = Number of activities in a nonsimultaneous set

J(S) = Number of possible sequences of those N nonsimultaneous activities when S is the maximum simultaneity

S = Maximum number of activities allowed in progress simultaneously

Any S - 1 of the N activities can be unrestricted by precedence if the remaining n - (S - 1) are made to form a single precedence path. Then only one of the N - S + 1 can be in progress at any one time and the S - 1 can be in progress simultaneously, giving a maximum simultaneity of S.

The number of ways in which such sequences can be formed is $N^P(N - S + 1)$, the total possible number of permutations of N items where each permutation contains N - S + 1 items. This number is $\prod_{i=0}^{N-S} (N - i)$. For example, consider a nonsimultaneous set which has four members, any two of which may be in progress at any one time. The above sequencing procedure would create (4)(3)(2) = 24 sequences.

In general, however, J(S) may be greater than $\prod_{i=0}^{N-S} (N - i)$ because other sequencing arrangements can be obtained to limit the simultaneity of the members of a set. For the example in the preceding paragraph (N = 4, S = 2) two sequences of two activities each also would limit simultaneity to the level two. The simultaneity limit achieved by sequencing is equal to the number of subsequences making up the sequence in question. As used herein, a subsequence may consist of a single activity.

The possible subsequences for a given N,S combination may be determined in the following fashion:

1. Each sequence will contain S subsequences.

2. One arrangement will have one activity in each of S - 1 subsequences and N - S + 1 activities in the S^{th} subsequence.

3. There are $N^P(N - S + 1)$ ways to design the S^{th} subsequence and for each of those ways, only one way to fill the 1^{st} through the

$(S - 1)^{st}$ subsequences. Note that subsequences are distinguishable only by size, not by position. Therefore the S - 1 subsequences of size one can be filled in only one way.

4. A second arrangement of subsequences allows the S^{th} subsequence to contain N - S activities. The $(S - 1)^{st}$ subsequence will contain two activities and the 1^{st} through the $(S - 2)^{th}$ subsequences each will contain one activity.

5. There are $N^{P}(N - S)$ ways to design the S^{th} subsequence, and for each of those subsequences there are $S^{P}2$ ways to design the $(S - 1)^{st}$ subsequence. For each of those ways to fill the S^{th} and $(S - 1)^{st}$ subsequences, there is only one way to fill the remaining single-member subsequences.

6. This procedure for distributing the N activities among S subsequences is continued until subsequence size combinations start repeating themselves. Table 2 illustrates the procedure for three N,S combinations.

Table 2. Examples of Subsequence Size Combinations for Varying N,S Values

	NUMBER IN SUBSEQUENCE								
	N=5 S=2		N=6 S=3			N=8 S=4			
	1	4	1	1	4	1	1	1	5
Subsequence	2	3	1	2	3	1	1	2	4
Size			2	2	2	1	1	3	3
Combinations						1	2	2	3
						2	2	2	2

7. The procedure for generating all subsequence size combinations may be described physically as:

(a) Forming S - 1 subsequences of size one and one subsequence of size N - S + 1.

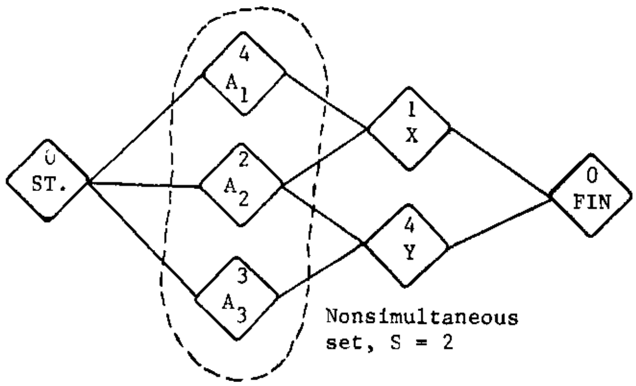
(b) Generating the succeeding size combinations by reducing the largest subsequence size by one activity and increasing by one activity the size of the largest subsequence which is two or more activities smaller than the one being reduced.

(c) Continuing the above procedure until there remains no subsequence smaller than one activity less than the size of the maximum subsequence, thus indicating that all size combinations have been generated. As there is no ordering among subsequences of a sequence, it is combinations of sizes, rather than permutations, which are generated.

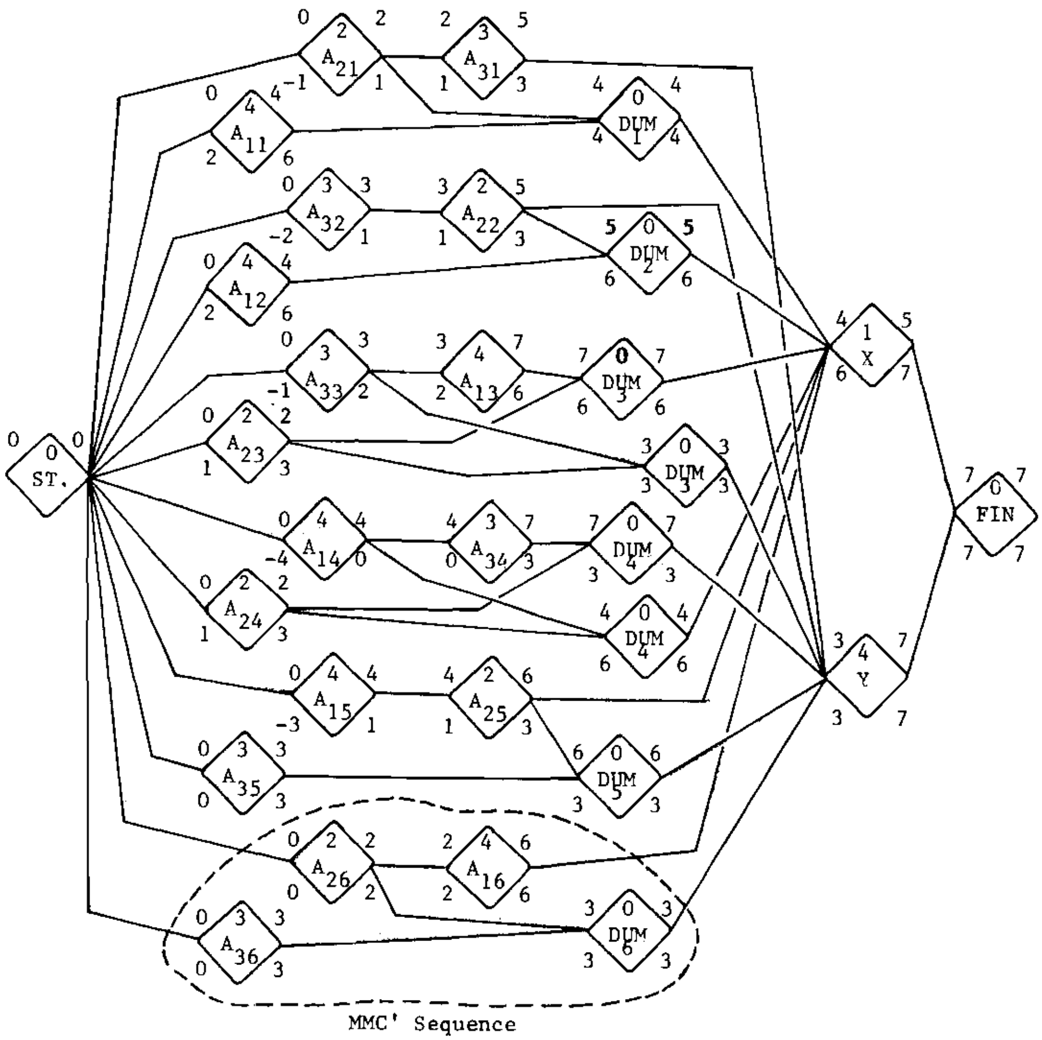
The number of sequences for a given subsequence size combination may be calculated. This number will be $\frac{N!}{k_1!k_2! \ldots k_{(N - S + 1)}!}$, where k_i equals the number of subsequences of size i in the subsequence size combination under consideration. For example, in Figure 15, on the following page, one of the size combinations for N = 8, S = 4, is 1, 1, 2, 4. There are $\frac{8!}{2!1!1!}$ sequences possible with that size combination because there are two subsequences of size one, one subsequence of size two, and one subsequence of size four.

Selecting Optimal Sequences

Given that the simultaneity maximum is greater than one, the calculation procedures for a forward pass must be amended. It is important that the ES' calculated for any activity be a feasible ES.



(a) Basic Network Containing Nonsimultaneous Set A_i



(b) Enumerative Network

Figure 15. Nonsimultaneity Graphics for Simultaneity Maximum Greater than One

By this is meant that there must be some sequence which will allow ES = ES'. To accomplish this, it becomes necessary that the first step in calculating ES' for a given activity, X, is to eliminate from consideration all but the $\max_i$ EC'(A_{iJ}), where the A_{iJ} >> X and J indicates one specific sequence of A_{ij}. Then the previously designed nonsimultaneity forward pass procedures hold.

The preceding modification may be handled graphically by having no more than one precedence arrow to activity X from any nonsimultaneity sequence. If X is a direct successor to more than one activity A_{iJ}, X is made an immediate successor to a dummy activity which is in turn made an immediate successor to the two or more activities A_{iJ}. A single such dummy will be associated with only one sequence, although there may be one or more dummies with each sequence.

Figure 15 illustrates the use of the dummy in a N = 3, S = 2 single-set nonsimultaneity problem. MMC' is associated with sequence A_{i6}. One can quickly verify that any other sequence would result in a greater project length than does sequence A_{i6}.

The nonsimultaneity procedures for single-set networks and independent multi-set networks are optimal for simultaneity maximums greater than one. It can be shown that all sequences for a given non-simultaneity set will have MC' = MC, just as for the simultaneity maximum of one. The MMC' sequence will have its maximum transfer of criticality to its successors be equal to MMC'. No other sequence can have its maximum transfer of criticality be less than MMC'.

The dependent multi-set network with simultaneity maximum greater than one presents formidable obstacles to attempts to prove that the nonsimultaneity procedures apply optimally to it. Limited experimentation has failed to find a nonoptimal example, however, so the procedures at this time may be said to give optimal answers with, at worst, encouraging frequency.

Implications of a Simultaneity Maximum Greater than One

It appears that the simultaneity maximum of more than one is more complex to deal with than the simultaneity maximum of one. A substantial part of this increased complexity is, however, in the creation of alternative sequences. This phase of the problem is common to both the complete enumeration approach and the single enumerative network procedures developed herein. When the simultaneity maximum increases above one, the computational problem is unchanged for the complete enumeration approach, but it is slightly increased for the enumerative network approach.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Summary

The nonsimultaneity constraint can be dealt with optimally in some of its more commonly occurring forms. The enumerative network approach provides optimal solutions in these cases with considerably less effort than is required by complete enumeration. Manual applications of the enumerative network approach are feasible in some problems of practical size. For repetitive use and large nonsimultaneity set size, computer implementation of the procedure appears highly desirable. The enumerative network approach offers the same magnitude of computational advantage when only a sample of the sequencing alternatives are to be examined.

The enumerative network approach possibly is optimal in those cases where the simultaneity maximum is greater than one. Further investigation is in order for these cases, however.

The theorems on transfer of criticality and consistency of EC' and ES' are additions to the state of the art in networking computation. Their use in other problem areas appears to be quite likely.

There are many resource allocation problems in the project management field. It appears that many of them might be structured as nonsimultaneity problems. A large number of specific topics for further investigation have come to light during this research. Many of these involve working to achieve greater generality in approaches to the nonsimultaneity problem. This greater generality will further open the resource allocation problem area to the nonsimultaneity approach.

Conclusions

The Nonsimultaneity Problem

The nonsimultaneity problem can be described explicitly. Its effects on network planning, scheduling, and control are of sufficient potential magnitude to warrant the design of a formal approach to dealing with this constraint. Optimal solutions to the nonsimultaneity problem are important for two primary reasons. One of these is the importance in some actual projects of minimizing project length. The other is the need for an optimal solution to use in evaluating the merit of simplified, but nonoptimal, nonsimultaneity procedures. The complexity of the nonsimultaneity constraint makes the intuitive approach have, except in the simplest cases, a negligible likelihood of achieving an optimal solution.

The Enumerative Network Procedure

The solution of nonsimultaneity problems using the enumerative network procedure is optimal for the following conditions:

1. Simultaneity maximum of one
2. Mutually exclusive sets
3. Single- or multi-set networks
4. Independent or dependent sets

The magnitude of the saving in computational effort of the enumerative network approach as compared to the complete enumeration approach is of very great practical significance. The example networks used herein have contained a minimum of activities other than those in the nonsimultaneous sets. In practice, the number of activities in nonsimultaneous sets would tend to be a very small percentage of the total number of activities. The computational advantages of the enumerative network procedures would be correspondingly more striking in the typical practical application.

Simultaneity maximums of more than one can be handled by the enumerative network approach. The results will not possess proven optimality, but are probably better than random selection of sequences.

Approaches to Implementation of the Nonsimultaneity Procedure

The network analyst is urged to consider whatever level of application of these techniques that applies to him. If one has a single $N = 3$, $S = 1$ network, the procedures of Chapter III may be applied manually with little difficulty. The approach of Chapter IV for independent sets may likewise be applied rather easily when multiple independent $N = 3$, $S = 1$ sets are in the network.

The manual approach is still quite feasible for $N = 3$, $S = 1$ when the dependent set procedure of Chapter IV is used. The value of N is not actually limited for manual processing, but $N = 5$ is a very difficult size to handle and $N = 6$ is quite unreasonable. Optimality is proven for $S = 1$ and not for $S > 1$, but the manual approach allows the processing of sets having $1 \leq S \leq (N - 1)$ with $N \leq 4$.

Until computer processing with the nonsimultaneity procedure becomes possible, the sets having $N = 5$ might be advantageously handled by a sampling procedure. As an example, one might use a Latin Square design to design eight sequences for a N = 8, S = 1 set. Instead of having to handle 8! sequences, which would be unmanageable, one sacrifices the assurance of optimality and is able to choose the best of the limited group of sequences generated. Recommendations concerning the sampling approach are made in later in this chapter.

If one wishes to use existing computer programs to process the network after the sequencing of a nonsimultaneous set, this can be done easily. There is no value to a full forward and backward pass made manually in this situation. The forward pass need go only through the sequences generated. If all paths following the sequences of a set pass through a common activity, the backward pass may be started at that activity and the LS may be an arbitrarily selected number. The information from the basic network can be amended to add the precedences associated with the MMC' sequence.

It is important that the user be aware that the optimum sequence needs to be redetermined at every updating. Deviations of actual progress from planned progress may cause a shift in criticality. This shift in criticality may reflected by a shift in the location of MMC'.

If a resource allocation feature is a part of the computer program being used, one may wish to try two alternatives. The first of these would be to deal with the nonsimultaneity constraint before submitting the network to the resource allocation procedure. The second might be to resolve the nonsimultaneity conflicts within the framework created by the resource allocation. A third possibility would be to carry out the nonsimultaneity procedures, both before and after the resource allocation.

It is anticipated that the identification of the nonsimultaneous sets will present some difficulties at first. It is suggested that activities which are actually nonsimultaneous, but whose feasible domains are widely separated, not be considered as having the nonsimultaneity constraint. A check of the final schedule to assure that such activities are not scheduled simultaneously will be faster than the handling of the additional sequences which would be required to handle the two activities in the recommended enumerative network approach.

Network Computation

The shortage of computational tools designed for network computation is retarding the production of significant advances in networking technology. The theorems on criticality are basic, but important. The theorem on EC' - ES' compatibility is more a special-purpose theorem. It is indicative of the type of network relationship which needs to be verified and documented. The techniques utilized in this paper for verbally describing precedence relationships allow for fairly concise description of conditions. In general, the descriptional and computational by-products of this research constitute a small but definite improvement in networking technology.

Recommendations

Proof of Optimality for Simultaneity Maximum Greater than One

The most pressing problem in the extension of the procedures designed herein is the lack of proof of optimality when these procedures are applied to nonsimultaneity sets having a simultaneity maximum of more than one. It appears that EC' - ES' compatibility does not exist, but that MMC' equality between sets may hold. It is suggested that initial efforts at proof of optimality seek to establish the existence of the MMC' equality.

Simultaneity Maximum Dependent upon Activities Involved

There are situations in which the number of activities from a given nonsimultaneity set which may be in progress simultaneously is not a constant. For example, if the set consists of A_1, A_2, and A_3, perhaps A_1 and A_2 may be in progress simultaneously. When A_3 is in progress, however, neither of the other activities may be in progress.

The variable simultaneity maximum arises when activities having varying resource quantity requirements are competing for a limited supply of that resource type. The practical effect of this facet of the nonsimultaneity problem is to create the need for a new approach to the design of feasible sequences. It is recommended that this problem be given high priority because of the widespread application an optimal solution would have.

Intersecting Nonsimultaneous Sets

There are situations in which two sets of activities, each having a nonsimultaneity constraint statement applying, are related through a third nonsimultaneity constraint. Figure 16, on the following page, illustrates this situation with sets A_i and B_i. Only one of the A_i may be in progress at any one time. Similarly, only one of the B_i may be in progress at any one time. The restriction that creates an intersection is the constraint statement that A_3 and B_3 cannot be in progress simultaneously.

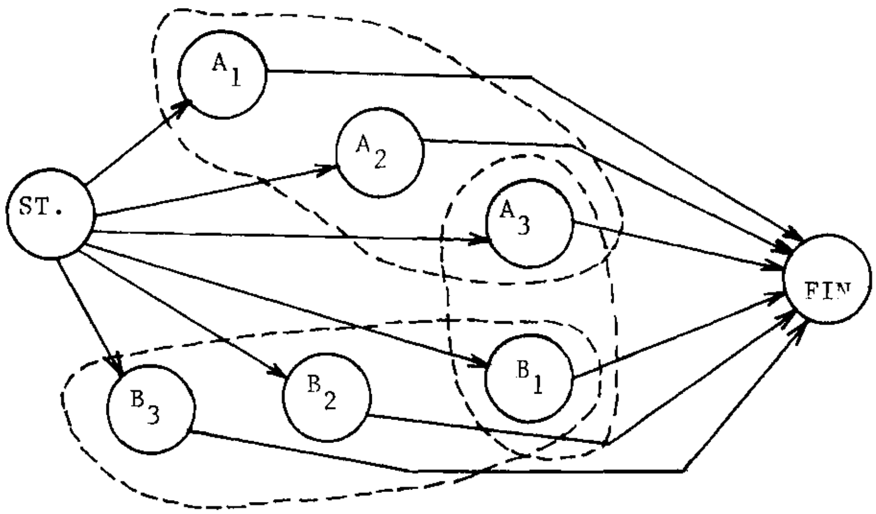


Figure 16. Intersecting Nonsimultaneous Sets

Sequence Sampling Possibilities

A nonsimultaneous set of N = 5 with S = 1 has 5! = 120 possible sequences. If a small sample of these 120 sequences were taken at random, the probability that it contained the optimal sequence would not in general be very large. It does seem, however, that the best sequence in the sample would have a high probability of having MC' not much larger than MMC' over all sequences.

It is recommended that investigation be undertaken to quantify the uncertainty inherent in such a sampling approach. The nonsimultaneity procedure can be used effectively in this study to give the MMC' to which MC' from the best sequence in the sample may be compared.

An advantage of sizable proportions may be gained if the configuration of sequences may be used to obtain at least a partial ordering of desirability. For example, in the applications studied, the optimal sequences tend to be those having the largest number of precedence connections. Other factors, such as ordering of durations, may be important in devising an improved sampling procedure. It appears possible that the one- and two-set network in general and the independent multi-set network might have optimal solutions available through a sequence configuration approach.

Precedence Connections among Members of a Set

The situation frequently arises in which two or more members of a nonsimultaneous set are precluded from simultaneity by the existence of a technological precedence constraint. This has the effect of reducing the number of sequences, because these activities cannot be permuted among themselves. A minor problem of sequence generation exists and merits limited study. A practical and more important aspect of this same problem arises in the evaluation of the effect of interruption of a series of similar activities by a member of the nonsimultaneous set to which the series belongs. The incorporation of such effects into the nonsimultaneity procedure would be of particular importance to the managers of construction projects, among others.

The Effects of Resource Limitations

The approach taken in this research considered all activities not in any nonsimultaneous set as being free to take place at any time consistent with the precedence requirements. However, limited resource availability may require that the project take longer than the length of the critical path on the modified network. If resource restriction is not handled by a nonsimultaneity approach, one may find that after allocating resources, some other choice of sequences would be better. More properly stated, nonsimultaneity resolution and resource allocation interact. Adaptation of the nonsimultaneity procedures to combine them with various resource allocation procedures is a challenging task. The resulting combination would, however, make both the nonsimultaneity procedure and the resource allocation procedure more useful.

Computer Processing

The routine for generation of alternative sequences becomes laborious as N becomes greater than three or four. The labor is highly repetitious and the data to be accounted for are extensive. The calculations involved are simple but numerous. These characteristics of the nonsimultaneity procedure make it important that a computer program be designed to take basic network data and nonsimultaneity information and generate at least the optimal sequencing. Ideally, the program also would carry out resource allocation, generating a calendar-dated schedule with a resource loading indication by time period. The program should have the capability of re-examining alternative sequences at each updating.

The importance of computer processing is becoming more pronounced with the advent of remote terminals on a time-sharing basis. The rapidity with which the project manager can specify the nonsimultaneity constraints and see their effects directly affects his tendency to explicitly identify nonsimultaneity problems.

Discounting the Effects of Future Decisions

Consider a set which will not be in progress until many time periods into the future. The selection of the optimal sequence for this set is of less importance than the selection of the optimal sequence for a set at the first of the network. A different sequence might be selected for the later set at each updating between the start of the project and the start of the first activity from this set.

The present worth of future value is a concept which might be employed to advantage in such situations. As an example, sets early in the network might be handled in the manner suggested for the enumerative network approach, with all possible sequences being generated. Sets later in the network might have only a sample of sequences generated, with the sample size based on the number of time periods from the start of the network. As the project progressed, a set would have more sequences generated until eventually all sequences were examined.

It is recommended that the possible value of discounting project management decisions about future problems be considered for research. Its use would not be restricted to nonsimultaneity work, but could be applied to resource allocation and time-cost trade-off as well.

The Effect of Unequal Direct Costs among Sequences

Situations arise in which a nonsimultaneous set is sequenced in a particular way because that sequence is found to involve less direct cost than the other possible sequences. It is more proper to evaluate such sequences by combining the direct cost of the sequence being examined with the indirect cost associated with selecting that sequence rather than the MMC' sequence. Explicit procedures for combining direct and indirect costs of alternative sequences need to be developed. Particular attention should be paid to the situation in which the MMC' sequence and one or more other sequences would not be on the critical path of the modified network.

Research Results

This research gives explicit definition to the nonsimultaneity problem. It summarizes the work of others on related problems. Specific procedures are described and proven optimal for some of the basic forms of the nonsimultaneity problem. Promising procedures are described for more complex nonsimultaneity problems. Tools and topics for future research also are given.

The most important result of this research is the improvement it has brought about in the writer's view of networking problems. The nonsimultaneity problem as originally conceived in this research resembles only slightly the multi-faceted problem eventually defined. The procedures adopted represent many cycles of design, testing, and modification. Much additional work remains to be done before general application of the nonsimultaneity concept is possible. This research, however, constitutes a significant first step and provides direction for subsequent research on this basic problem.

BIBLIOGRAPHY

1. Baumgartner, John S., *Project Management,* Richard D. Irwin, Inc., Homewood, Ill., 1963.

2. Burgess, A. R., and James B. Killebrew, "Variation in Activity Level on a Cyclical Arrow Diagram," *Journal of Industrial Engineering,* March-April, 1962, pp. 76-83.

3. Clark, Charles E., "The Optimum Allocation of Resources Among the Activities of a Network," *Journal of Industrial Engineering,* January-February, 1961, pp. 11-17.

4. Clark, Wallace, *The Gantt Chart,* Ronald Press Co., New York, 1922.

5. *DOD/NASA Guide to PERT/COST Systems Design,* Office of the Secretary of Defense and the National Aeronautics and Space Administration, June, 1962.

6. Eisner, Howard, "A Generalized Network Approach to the Planning and Scheduling of a Research Project," *Operations Research,* Vol. 10, 1962, pp. 115-125.

7. Frambes, Roland, "Next Big Step for PERT," *Aerospace Management,* October, 1961, pp. 77-78.

8. Kelley, J. E., Jr., "The Critical Path Method: Resources Planning and Scheduling," a paper presented at the Factory Scheduling Conference, Carnegie Institute of Technology, May 10-12, 1961.

9. Kelley, James E., Jr., "Critical Path Planning and Scheduling: Mathematical Basis," *Operations Research,* Vol. 9, 1961, pp. 296-320.

10. Knoeppel, C. E., *Installing Efficiency Methods*, The Engineering Magazine, New York, 1915.

11. Koepke, Charles A., *Plant Production Control,* Second Edition, John Wiley and Sons, Inc., New York, 1958.

12. Levy, F. K., G. L. Thompson, and J. D. Wiest, "Mathematical Basis of the Critical Path Method," a paper published in *Industrial Scheduling,* edited by John F. Muth and Gerald L. Thompson, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1963.

13. *Line of Balance Technology,* Office of Naval Material, Department of the Navy, NAVEXOS P1851 (Reviewed and Approved April 17, 1962).

14. Malcolm, D. G., "Reliability Maturity Index (RMI) - An Extension of PERT into Reliability Management," *Journal of Industrial Engineering,* January-February, 1963, pp. 3-12.

15. Malcolm, D. G., J. H. Roseboom, C. E. Clark, and W. Fazar, "Application of a Technique for Research and Development Program Evaluation," *Operations Research,* Vol. 7, 1959, pp. 646-669.

16. Martino, R. L., *Project Management and Control, Volume III--Allocating and Scheduling Resources,* American Management Association, New York, 1965.

17. Mauchly, John W., "Critical Path Scheduling," *Chemical Engineering,* April 16, 1962, pp. 139-154.

18. Moder, Joseph J., and Cecil R. Phillips, *Project Management with CPM and PERT,* Reinhold Publishing Corporation, New York, 1964.

19. Moshman, Jack, Jacob Johnson, and Madalyn Larsen, "RAMPS, a Technique for Resource Allocation and Multi-Project Scheduling," *Proceedings, 1963 Spring Joint Computer Conference,* pp. 17-27.

20. Muth, John F., and G. L. Thompson, *Industrial Scheduling,* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1963.

21. *RAMPS Users Guide,* C-E-I-R, Inc., Washington, D. C., 1963.

22. Schaffer, L. R., J. B. Ritter, and W. L. Meyer, *The Critical Path Method,* McGraw-Hill Book Company, Inc., New York, 1965.

23. Taylor, Frederick W., *Scientific Management,* Harper and Brothers, New York, 1947.

24. Verhines, D. R., "Optimum Scheduling of Limited Resources," *Chemical Engineering Progress,* March, 1963, pp. 65-67.

25. Walker, M. R., and J. S. Sayer, "Project Planning and Scheduling," Report 6959, E. I. du Pont de Nemours and Company, Inc., Wilmington, Delaware, March, 1959.

VITA

Johnny Gordon Davis was born November 9, 1933, at Perry, Florida, the son of John Hancock and Mary Myrtle (née Griffin) Davis. He attended public schools in Perry, graduating from Taylor County High School in June, 1951.

He entered the University of Florida in September, 1951, as a Lewis Scholarship student. Mr. Davis became a member of Lambda Chi Alpha social fraternity in November, 1952. He attended the University of Florida until January, 1954, at which time he entered the construction industry. On December 16, 1955, he was married to Billie Joan Tillman of Americus, Georgia.

Mr. Davis entered the United States Army in March, 1956, and served as a cryptographic crew chief in the Signal Corps. On January 11, 1957, a son, Johnny Michael, was born.

In February, 1958, Mr. Davis was released from active duty to resume his studies at the University of Florida. During his junior and senior years he served as vice-president of the student chapter of the American Institute of Industrial Engineers, and secretary of the Benton Engineering Council. He was elected to membership in Sigma Tau. He graduated in January, 1960, with the degree Bachelor of Industrial Engineering with Honors.

Mr. Davis entered the Graduate School of the University of Florida in February, 1960. He served as a Research Assistant in the Department of Industrial Engineering and later as a Teaching Assistant in the Engineering Graphics Department. His Master's research resulted in a thesis entitled "A Model for Improvement of Hospital Operating Room Utilization." He graduated in June, 1961, with the degree Master of Science in Engineering.

In June, 1961, Mr. Davis joined the faculty of the School of Industrial Engineering of the Georgia Institute of Technology as an Instructor. On July 20, 1962, a son, Thomas Allen, was born. Mr. Davis resigned from the faculty in September, 1962, to become a Ford Fellow in the School of Industrial Engineering. He attended Georgia Tech until June, 1964, on the Ford Program, and from June, 1964, to September, 1964, he attended as the holder of a National Science Foundation Fellowship. He returned to the faculty as an Assistant Professor in September, 1964. On September 6, 1966, a third son, Griffin Lee, was born.

In addition to his teaching in the School of Industrial Engineering, Mr. Davis has taught and administered courses in Georgia Tech's Department of Continuing Education and has served as a consultant to many industrial firms. He is a member of the American Society for Engineering Education and the American Institute of Industrial Engineers. He is co-author of the article, "Variability Control is the Key to Maximum Operating Room Utilization," published in *The Modern Hospital*, April, 1964.